Informatics

Auf dem Weg zu einem generalisierten Reason Model

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Logic and Computation

eingereicht von

Henri Thölke, B.Sc.
Matrikelnummer 12223171

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ. Prof. Dr. Agata Ciabattoni

Wien, 20. August 2024

Henri Thölke

Agata Ciabattoni

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Towards a generalized Reason Model

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Logic and Computation

by

Henri Thölke, B.Sc.
Registration Number 12223171

to the Faculty of Informatics

at the TU Wien

Advisor: Univ. Prof. Dr. Agata Ciabattoni

Vienna, August 20, 2024

Henri Thölke

Agata Ciabattoni

Erklärung zur Verfassung der Arbeit

Henri Thölke, B.Sc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 20. August 2024

Henri Thölke

Danksagung

Zunächst möchte ich meiner Betreuerin Univ. Prof. Dr. Agata Ciabattoni danken. Ich danke Ihr für die schnellen und umfangreichen Rückmeldungen zu meinen Fragen und Ideen, und dafür dass sie mich stets herausgefordert hat das Beste aus meiner Arbeit rauszuholen. Insbesondere möchte ich mich bedanken für die Ermutigungen und die Angebote mich intensiver mit dem akademischen Umfeld und der Forschung zu beschäftigen.

Ebenso danke ich Univ. Prof. Dr. Laura Kovacs und Associate Prof. Dr. Magdalena Ortiz für meine Anstellung als Tutor. Ich danke Univ.Prof. Dr. Hans Tompits für viele spannende Lehrveranstaltungen die mir einen Einblick über viele Bereiche der Logik ermöglicht haben.

Ich danke Ilaria Canavotto und Xinghan Liu für ihre sehr freundliche Hilfe und ihr Feedback zu meiner Arbeit an dieser Thesis. Die Diskussionen ermöglichten es mir, ein besseres Verständnis der aktuellen Forschung zu erlangen.

Schließlich möchte ich meiner Familie danken. Ich danke meinen Eltern für die unglaubliche Unterstützung während meines gesamten Studiums und insbesondere bei der Entscheidung das Studium in Wien anzustreben. Ich danke Alessia Marchetti für alle Tage die wir zusammen verbracht haben und die wir in Zukunft zusammen verbringen werden, sowohl für die produktiven als auch für die entspannten. Ich danke meinem Bruder, meiner Schwester, meinen Freundinnen und Freunden für die Gespräche und guten Zeiten zusammen.

Acknowledgements

First I would like to thank my advisor Univ. Prof. Dr. Agata Ciabattoni. I thank her for the quick and comprehensive feedback to all my questions and ideas, and for challenging me to make the most of my thesis at every step. In particular I would like to thank her for the encouragement and the offers to engage more intensively with the academic environment and research.

I also thank Univ. Prof. Dr. Laura Kovacs and Associate Prof. Dr. Magdalena Ortiz for my employment as a tutor. I thank Univ.Prof. Dr. Hans Tompits for many engaging Lectures that allowed me to gain a perspective over many aspects of logic.

I thank Ilaria Canavotto and Xinghan Liu for their very friendly help and feedback throughout my work on this thesis. The discussions allowed me to get a better understanding of the state of current research.

Finally I would like to thank my family. I thank my parents for the unbelievable support throughout my entire studies, and in particular with the decision to pursue studying in Vienna. I thank Alessia Marchetti for every day we spent together already and every day we will spend together in the future, the productive ones as well as the relaxed ones. I thank my brother, my sister and all of my friends for the conversations and good times together. You all were a big reason why these last years were so amazing, so thank you very much.

Kurzfassung

Normative reasoning ist die Untersuchung von Normen mit logischen Methoden, und wird in einer Vielzahl von Anwendungen eingesetzt, da Normen ethische, rechtliche, soziale oder kulturelle Normen beinhalten. Ein Weg, in dem Normen in der Informatik eingesetzt werden ist um das Verhalten von künstlicher Intelligenz (KI) zu spezifizieren und zu steuern. In diesem Kontext ist eines der offenen Probleme das der Norm Akquirierung. Der aktuelle Stand der Technik besteht aus zwei Ansätzen: Einer nutzt deklarative Repräsentationen mithilfe einer logischen Sprache, welche üblicherweise per Hand erstellt wird. Der andere Ansatz nutzt Machine Learning um Normen aus Trainingsdaten zu erkennen und zu lernen. Beiden Ansätzen fehlen einige gewünschte Eigenschaften wie die Möglichkeit der Skalierung oder Transparenz.

Die natürliche Idee ist daher einen Hybridansatz zu untersuchen, welcher die beiden bisherigen Ansätze vereint und ihre Stärken und Schwächen miteinander ausgleicht. Ein Framework welches für einen solchen Ansatz genutzt werden könnte sind Modelle zum Argumentieren über Präzedenz, welche für rechtliche Argumentation entwickelt wurden. Insbesondere das *Reason Model* welches von John Horty entwickelt wurde stellt sich als passender Kandidat heraus. Das Hauptziel der Entwicklungen um das Modell ist es von reinem rechtlichen Argumentieren zu lösen und es einem allgemeineren Kontext von KI zu öffnen. Eine dieser Entwicklungen ist zum Beispiel die Kodierung des Modells in einer Modallogik, welche genutzt werden kann um Erklärungen zu generieren.

Diese Arbeit ist ein Schritt hin zu einer Generalisierung des Reason Models für die Nutzung in KI. Das Langzeitziel dieser Thesis ist die Entwicklung eines Frameworks welches eine transparente, verifizierbare und erklärbare Representation von normativer Information bietet, die automatisiert akquiriert werden kann.

Abstract

Normative reasoning is the logical study of norms, a concept which is found in a multitude of domains, as norms range from from ethical, legal, social or cultural norms. In computer science, one way norms are used is to specify and regulate the behavior of artificial intelligence (AI) systems. In this context, one of the open problems is that of norm acquisition. The current state of the art consists of two approaches: One uses a declarative representation using a logical language, which is typically encoded by hand. The other uses machine learning to detect and learn norms from training data. Both approaches lack some desired properties like scalability or transparency.

The natural idea is therefore to investigate a hybrid approach that combines the two and balances out their strengths and weaknesses. One framework that could be used for such an approach are precedential reasoning models that are developed for legal reasoning. In particular the *Reason Model* developed by John Horty turns out to be a suitable candidate. The main goal of the developments surrounding the model is to move it away from purely legal reasoning, opening it up to a general application in AI. One of these developments for example is the encoding of the model in a modal logic, which can be used to generate explanations.

This work is a step towards generalizing the reason model for the use in AI. The long term goal of this thesis is to develop a framework that provides a transparent, verifiable and explainable representation of normative information that can be acquired automatically.

Contents

CHAPTER 1

Introduction

Norms, which include obligations, permissions or prohibitions, have long been investigated in many fields of study. In philosophy, ethical norms describe a framework of what may constitute ethical conduct. In law, norms describe the rules or concepts at the base of the laws built upon them, and social norms describe what is considered acceptable behavior in society. A discipline that connects to all these applications of norms is computer science, where norms are used for example to regulate the behavior of artificial intelligence (AI). This application is motivated by the growing presence of AI systems in human life, which means that aside from showing impressive results, these systems must also have a significant level of normative competence. Naturally, an AI that is to be used in the courtroom must follow legal norms, an AI surgeon must follow the norms of medical practice and so on.

Norms can be distinguished from traditional rules or laws by a few key characteristics. Norms often have many exceptions for specific circumstances, as well as open formulations that require additional interpretation. On top of that, the nature of norms is that they can be violated, something which is not the case for the laws of physics for example. The logical study of norms, called *Normative Reasoning*, therefore differs noticeably from factual reasoning.

The use of normative reasoning in AI is a part of *Machine Ethics*, a research field that aims to build AI systems that act ethically, where it is used to enable AI systems to follow given norms. This development includes AI systems for a variety of domains where norms appear, from ethical and legal norms, to societal or cultural norms.

One open problem of using norms to regulate AI systems is that of knowledge acquisition. Normative information is not easily measurable like factual data and we therefore need different methods to acquire norms, and to represent them in a way the machine can use. Furthermore, the normative information needs to be transparent and verifiable. Without a way to understand or explain its decisions, humans will hardly trust or accept the correctness of an AI systems decisions in highly impactful settings like for example the courtroom or the hospital. Representing norms in an understandable and verifiable way is therefore essential if AI systems are to be used in these settings.

There are currently two approaches on how to acquire and represent normative information. The *top-down* approach provides the AI system with explicit normative information, using a declarative representation of knowledge. Decisions are then reached using logical reasoning mechanisms, often connected to a precise semantic theory. On the other side, the *bottom-up* approach is using machine learning. The system uses large amounts of data to learn normative information through identifying patterns in the training data. The knowledge is stored in machine learning models like neural networks.

Both the top-down and the bottom-up approach have clear advantages, as well as shortcomings. Notably, the approaches have complementary strengths and weaknesses. While the top-down approach results in transparent and explainable representations of knowledge and reasoning as a result of a symbolic and declarative representation and clearly defined semantics, the process of encoding the normative information in the chosen representation is typically done manually, which makes the acquisition of large and complex normative systems very impractical.
On the other hand, the machine learning models used for the bottom-up approach can be trained easily[1], however their behavior is not transparent. Machine learning system essentially look for statistical patterns in the encoded training data, removing the semantics of the data entirely from the process. This makes the quality of the training data immensely important for the behavior of the system [JPN$^+$20][2].

Naturally then, a hybrid approach comes to mind, one that harnesses the capabilities of machine learning to efficiently acquire normative information at scale, but that uses a symbolic representation that allows for human-understandable and explainable reasoning. This thesis contributes to the research into such an approach.

One framework that has been considered a candidate for this hybrid approach comes from the domain of automated legal reasoning. In particular, the model uses *Case-Based Reasoning*. In case-based law, the relevant legal norms are established not through statutory law (that is written, explicit law) but instead through the decisions that courts make in individual cases. The guiding principle in case-based law is that of precedent. Precedent is the legal concept that past decisions of the courts impose a certain bind on present courts to decide new cases consistently with past decisions. In practice, this form

[1] Sometimes even using natural language, like in [RH16] where the authors propose using stories written in natural language to teach values to AI agents.

[2] For an example of problems that arise from this, see some research into how machine learning systems may be biased [MMS$^+$21].

of legal reasoning is applied in *common law*.
The principle of precedent is clearly a very intuitive guiding principle for normative reasoning, and so using it as inspiration for a generalized model is quite natural. Additionally, the field of automated precedential reasoning in the legal domain has been explored extensively in the last decades [RA87, AA97, Hor11, HB12, Pra21].

The formal framework that will be the basis of this work is one of these models for precedential reasoning, specifically the *Reason Model* introduced by Horty in [Hor11, HB12] and some of the modifications published in recent years [Pra21, Hor19, VGPV23]. These precedential reasoning models consist of a case base and a reasoning mechanism that defines how precedent cases influence the courts decision of new situations. Aside from being a well established framework in legal reasoning, choosing the reason model as our candidate for a hybrid approach is further motivated by a paper from Horty and Canavotto [CH22] in which they argue for using the reason model in applications outside of the legal domain. In their paper, the authors present the reason model in two non-legal contexts, as a decision procedure in an automated child-care robot and for deciding between possible donor organ recipients. They argue that using a case-based representation along with the principle of precedent offers a solution to the acquisition and representation problem of normative information. The envisioned system would first gather some case information, either by observing humans or by extracting facts from written documents using machine learning. The information would be encoded in some logical representation, which can then be provided to a reasoner that applies precedential reasoning to derive constraints that guide the decisions for new situations.
Another recent development related to the reason model comes from Lorini et al. who use a logic developed for classifiers to encode the reason model [LLRS22]. This work represents another connection of the reason model to AI, as classifiers are a very common example of AI systems. In general, a classifier is a system that takes as input a set of features and outputs a classification of those input features. Examples include sentiment analysis in natural language processing [SSS17] where the input is natural language and the system classifies text into a category based on sentiment, or classifying cancers in medicine [MLB^{+}23]. The observation of the authors of [LLRS22] is that the reason model acts exactly like a binary classifier. Each case contains a set of features, and the outcome is a decision in favor of one of two sides, plaintiff or defendant. Interpreting the reason model in this way opens it up to the existing research into classifiers, and provides a more formalized setting, aka a formal logic. Most importantly, this approach allows the authors to obtain explanations of the decisions of the model, which as we pointed out, is an important step towards trust and acceptance of these AI systems.

Both these papers show the interest in connecting the reason model and similar approaches to AI and using them to acquire normative information and reason with it. Horty and Canavotto provide a strong argument for their concept: The normative information is represented in the decisions of the individual cases, and so it is easily verifiable and understandable. Using precedential reasoning provides an intuitive mechanism to inform

or constrain future decisions, and since each individual case does not require a complex representation, acquiring the case information could realistically be automated. The results of [LLRS22] with their encoding of the model into a logic then allow for providing explanations of the decisions.
However, the very simple framework that the reason model presents means that its expressive power is very limited. For the use outside of the context of legal reasoning, e.g. in AI, further properties or features are needed. We will see that a lot of these features have been developed independently already, and our goal is to combine some of them into one single model as a first step towards a model suitable for the use in AI.

We will focus on four such properties that are desirable for a more generally applicable model.

1. A rich representation of information Naturally, the case information that is relevant for the decision needs to be adequately represented. The main challenge when representing the case information lies in the trade-off between detail and abstraction. The more abstract the information is represented, the easier the reasoning can be. Just think of propositional logic, which removes basically all context from a proposition, which allows for very simple syntax and semantics. However, a purely propositional representation might remove too much of the information, failing to capture relevant nuances between cases.

2. An intuitive and expressive reasoning mechanism The intuition of case-based reasoning is that similar cases from the past can inform new cases that our model is presented with. There are two measures that the reasoning mechanism must balance: First, what does it mean for a past case to be similar to some new case, and second how exactly does a past decision influence a current one. For an extreme example, consider a system that simply checks if a new case is exactly like some old case, and if this is true, the new case needs to be decided the same way as the old case. Such a system clearly provides no actual benefit. On the other hand, broadening the understanding of what is relevant too much not only complicates the reasoning, but it may also introduce unwanted effects, where unrelated information from a past case is used to influence the current decision. Similarly, increasing the amount of influence the old cases have reduces the ability of the system to adapt and lowers its flexibility. Alongside these considerations is the permanent demand for the reasoning principles to be intuitive and verifiable. The reasoning mechanism we choose will need to try to strike this balance.

3. A nuanced result The original reason model is used to constrain the decision of a court on whether to rule for the plaintiff or the defendant in a given case. For more general scenarios, being strictly limited to two outcomes is not sufficient however. This is because in a general setting, there is an important difference between one side barely winning out and it being clearly favored. This is for example needed when the model is used to make a choice between the lesser of two evils, or to chose the optimal of many good solutions. If all bad outcomes are simply labeled bad, and all good outcomes labeled good, then these choices are impossible[3].

4. Dealing with conflicting precedents One of the biggest assumptions made by most of the precedential reasoning models in the literature is that the case base of precedent decisions contains to contradictions or conflicts. This assumption however is very strong and impractical, as it is highly unlikely even for legal case bases to contain absolutely no conflicts. When we move to more general contexts, it will be even more likely that the case bases are in some way inconsistent. Consider the child care robot we mentioned above. If the case base contains decisions of parents from multiple families, it is very unlikely that they consistently agree on every situation. Our model needs to be able to resolve these conflicts, and obtain meaningful answers on how case bases that contain conflicts can constrain future decisions.

The contribution of this thesis will be the definition of a new model guided by these four goals, as well as a formal logic that builds off the classifier logic in [LL23] to encode more precedential reasoning models, all of which should be a step towards the development of a framework for the use in general AI applications. As we follow the ideas of frameworks developed for legal reasoning, the thesis is an interdisciplinary work, spanning legal reasoning, logic and AI. The thesis is organized as follows. In Chapter 2 we will provide a formal basis, as well as present a selection of precedential reasoning models in more detail, as we will draw from their properties to define our new model. We will also show the connections of these models to formal logic, which includes the encoding of the reason model into a modal logic for classifiers. Following that, in Chapter 3 we present the new contributions of this thesis. The first is a formal comparison between two of the reasoning models. The second is the introduction of a new reasoning model that aims to combine some relevant aspects of previous models to move towards a reasoning model for more general AI applications. The third contribution is a generalization of the model to enable it to reason based on inconsistent case bases. Finally, we define a new classifier logic and encode one of the modifications of the reason model using it. We give an overview of related work in Chapter 4 before summarizing our work in Chapter 5 and providing an outlook on future areas of research.

[3] In the context of normative reasoning, dealing with these situations is an important quality. Different deontic logics for example offer a variety of solutions to this problem, like a preference relation of possible worlds in dyadic deontic logic [Åqv02].

CHAPTER 2

State of the art

Models of precedential reasoning are the foundation for this work. The new model we present in Chapter 3 is a combination and generalization of models that have been developed for automated legal reasoning over the last years [HB12, Hor20, Pra21, VGPV23]. Therefore we present the models that we base our new model on in detail in this section. While we focus on the technical aspects of the models, it is important to point out the intuitions behind each modification. As the models have been developed for legal reasoning, a lot of the motivation for the modifications comes from legal theory, but just as we have argued that the basic principle of precedent carries over to a more general context, so do other aspects. Based on these intuitions we can identify which elements will give us the desired properties, and thus which elements we need to carry over into our new model for general AI applications.
Additionally, we present two connections of the original reason model of precedential constraint to formal logic. The first such connection comes in the form of the deontic logic created by the reason model. Deontic logics are a part of normative reasoning, and can help to solve some common issues of norms, like conflicting norms, or situations where norms are violated.
The second connection is that to binary classifiers. As mentioned in Chapter 1, in their recent paper [LLRS22] Lorini et al. use a modal logic developed for binary classifiers to encode the reason model of precedential constraint.

We begin by providing definitions for the important basic concepts of precedential reasoning in Section 2.1, as well as fixing some notation. This includes how precedential reasoning models are structured, and the most basic form of such a model. Then we will introduce the setting of our running example in Section 2.2. We will use the setting of this running example to show how the different models implement precedential constraint, and what results they provide for certain situations. Sections 2.3 to 2.7 then each introduce a model for reasoning with precedents. We present the original reason model, followed by a small generalization. Following that, we present two modifications that introduce a new representation of case facts and an adaptation of a third model that introduces a hierarchy to the case facts. We conclude the chapter in Section 2.8 with the connections of the reason model to formal logic.

2.1 Preliminaries of precedential reasoning

All of the models presented in the following sections are models for precedential reasoning in the legal context, and thus use legal terminology. We will adopt the language of these papers for simplicity. We will see in our running example how the legal terminology may correspond to other domains.
The basic scenario that all the models we discuss are based on is that of a court case: A *court* is presented with a case, and has to make a ruling for one of two sides, either the *plaintiff* or the *defendant*. The courts decision is restricted only by past cases that dealt with similar situations, called *precedent cases*. These restrict the court by requiring it to stay consistent with past decisions. The two main considerations of this restriction are which precedent cases are relevant for the new case and how consistency with past decisions is defined. These two considerations determine the precedential constraint that the model implements.

To investigate these models, it is important to understand their basic structure. From a theoretical standpoint, all the models we discuss are knowledge-based systems, and as such they usually consists of two separate components:

1. A knowledge base of cases.
2. A reasoning mechanism to constrain future decisions.

When we present a precedential reasoning model, we will always follow the same template, which is based on this structure. In the beginning we will lay out the intuitions of the models properties, to understand the motivation of that particular model. Then we will present the way the model defines its cases, that means how the case knowledge is represented. Finally, we present the definition of precedential constraint that the model provides. This definition will formally express the intuition we outlined in the beginning. Throughout the presentation we will provide examples to better illustrate the concepts.

In this section we present the most basic model for precedential reasoning, which uses a basic representation of a case, along with a very simple reasoning mechanism, namely *a fortiori* reasoning. We use the notation of [Hor11], and use it to define the model the author calls the *result model* of precedential reasoning.

The intuition of this most basic model is really to be just that, a simple and foundational model. This is particularly evident in the form of constraint it implements, that is *a fortiori* constraint. A fortiori reasoning is reasoning based on the strength of an argument. In the context of precedent, this means that if a new case presents a strictly stronger argument for a given side than a precedent case, then the new case must be decided the same way as the precedent case. The entire goal of the model is therefore to capture a formal meaning of the "strength of an argument for a side". To see how this is achieved in Horty's result model of precedential reasoning, we now begin with the knowledge representation.

Knowledge representation We will now introduce the basic representation of cases, starting with the *fact situation*. This fact situation describes all the relevant information that the court needs in order to make its decision. What the fact situation contains will vary depending on the framework we discuss, here we will begin with the simplest form, a fact situation based on factors[1].

Definition 1 (Factor, Fact situation)**.** A factor is a pattern of facts that is legally relevant. A fact situation is a set of factors.

We will denote fact situations using capital letters. A factor is denoted by a lower case name, $f_1, f_2, \dots$ for generic factors, or some telling name in examples. One crucial aspect of these factors is that each factor always favors exactly one and always the same side of a case. That means that if a factor is present in some situation then the facts that the factor stands for are an argument for exactly one side of the case.

The question which sides a factor can favor leads us directly to the second component of our case representation, the *outcome* that the court decided for. In the basic models this outcome is a decision either for the plaintiff or the defendant. We will use π to refer to the plaintiff, δ for the defendant, s for an arbitrary outcome and $\overline{s}$ for the outcome opposite to s.

We can thus split the set of all factors we consider into two subsets, those factors that favor the plaintiff and those that favor the defendant. For a given fact situation F we denote the subset of factors that favor some outcome s by F^s. We can therefore express a fact situation F as the union of factors for the plaintiff, denoted F^π, with the factors for the defendant, denoted F^δ:

$$F = F^\pi \cup F^\delta \text{ with } F^\pi \cap F^\delta = \emptyset$$

The intersection of F^π and F^δ being empty reflects the stipulation that each factor can only ever favor one side and never both. For an individual factor f_1 we will indicate that it favors side s by writing f_1^s.

[1] Using factors to represent legal cases is a subject of formal legal reasoning explored by many researchers. Most notably, two legal reasoning projects, HYPO [RA87] and CATO [AA97], use some form of factor based representation. The factors we present here are used as they are in the literature, without considering the issues with the formalism pointed out in the past, like identifying what constitutes a factor or how to assign polarities.

Finally then, we can formally define the notion of a *case* in its most basic form.

Definition 2 (Case)**.** A case $c = \langle F, s \rangle$ consists of a fact situation F and an outcome s where s is either π or δ.

The knowledge base that makes up one part of our precedential reasoning model and that contains all precedent cases is then accordingly called a *case base*.

Definition 3 (Case base)**.** A case base is a set Γ of cases.

Having defined this basic case representation, we can then look at the second component, the precedential constraint that the model defines.

Constraint As we mentioned above, the constraint of this basic model we present is a fortiori constraint. Remember that a fortiori reasoning means reasoning based only on the strength of an argument for a side. In the context of our case definition, that means that the reasoning is based on the strength of the argument contained in the fact situation for one of the two outcomes. If the new fact situation that is presented to the court makes a stronger argument for one outcome than a precedent case with that outcome, then the court must decide that new fact situation the same way that the precedent case was decided. This naturally requires some formal notion of strength of a case for an outcome. However, the factor-based representation we defined allows for a very simple and intuitive definition of strength for an outcome.

Definition 4 (Strength of a fact situation for an outcome)**.** A case $b = \langle F_b, s \rangle$ is at least as strong for outcome s as case $c = \langle F_c, s \rangle$ if and only if its fact situation F_b contains all the factors in F_c that favor outcome s, and no factor favoring the opposite side $\overline{s}$ that is not also contained in F_c. Formally, that means b is at least as strong as c if and only if

$$F_c^s \subseteq F_b^s \text{ and } F_b^{\overline{s}} \subseteq F_c^{\overline{s}}$$

We can then use this definition of strength of an argument to define the a fortiori constraint of the result model and obtain our first model for precedential reasoning.

Definition 5 (Constraint, result model)**.** Given a case base Γ and a new fact situation F, the court is forced to decide F for side s if and only if there exists a case $c \in \Gamma$ such that F is at least as strong for outcome s than c.
The court is permitted to decide F for side s if and only if it is not forced to decide F for $\overline{s}$.

This concludes the most basic aspects of representation and constraint in precedential reasoning. We will introduce changes or adaptations in the beginning of each section that introduces a model. First however, we will provide a scenario for our running example and use it to show how the result model can be applied to it. We will also use the running example to shows how to apply the concepts outside of the legal context.

2.2 Running Example: Dog adoption

Since the long-term goal that this work contributes to is to build normative AI systems for a broad variety of contexts, we will use a running example that applies the precedential reasoning models in a non-legal context. Specifically, we will use the process of deciding dog adoption applications. In the scenario, a dog shelter wants to use an automated system to decide whether an applicant will be allowed to adopt a dog. This scenario describes a situation which does require some amount of normative competence of the AI system, as pet adoptions carry some ethical significance. We would therefore want a system that justifies its decisions and is consistent. Additionally, the scenario provides us with a clear example of how the knowledge acquisition could work in practice. The system could be provided with the data taken from past applications, a task that can easily be automated, along with the decisions in each case. Using the representation of the respective model, the system is then able to derive the complex norms expressed in the decisions.

To fix the terminology, we translate the legal terms to our scenario. The court would be the shelter, with the two outcomes being a decision for or against the applicant. We will say that a decision for the plaintiff π is the outcome that the application is granted, and the applicant gets to adopt a dog, and a decision for the defendant δ is the outcome that the application has been denied.

We begin by introducing a basic scenario that uses the representation described in the previous Section 2.1. Whenever a model changes the representation, we will adapt the examples accordingly.

In our scenario, the shelter decides applications for adopting dogs based on factors that are obtained from the application form. These factors include whether an applicant has sufficient income to afford caring for a dog, whether the applicant has a living situation with the space necessary for a dog, whether the applicant has previously owned a pet, and whether they work in shifts.

We will denote these factors using the names

$$suffInc, smallApp, prevPet, shifts$$

with the factors $suffInc$ and $prevPet$ favoring the plaintiff, and the factors $smallApp$ and $shifts$ favoring the defendant[2]. Using this scenario and the result model for precedential reasoning we presented above, we can now make a first example, that shows the result model in action.

Example 1. Since we have the scenario established, we can directly start by considering a case base that contains just one case, that of Hans. To represent his case, we just need to provide the fact situation and the decision taken by the shelter. For Hans, we have

[2] One big aspect of factor-based case analysis is dealing with negation. For our examples, we will strictly follow the formalisms and avoid any debate on this subject, however it is clear that dealing with present, absent, or negated factors requires some additional thought. For a brief view on the topic, see Chapter 4 Section 4.3

the fact situation

$$F_h := \{suffInc, smallApp\}$$

and we suppose that the court decided to rule in in favor of Hans.
This means that we have the case

$$h := \langle F_h, \delta \rangle$$

in our case base Γ. Now to see how we apply our reasoning models, we always need to consider a new fact situation, and then apply the precedential constraint of the model we are considering to determine what decisions the court may take.

For this example, we consider the new fact situation of Isabelle:

$$F_i := \{suffInc\}$$

To see how precedential constraint influences the courts decision, we can then ask whether the court is permitted to rule for either outcome, or whether the court is forced to rule for one of the two. In this case, the answer is that the court is forced to rule in favor of Isabelle according to the result model. This is the case because her fact situation contains at least all the pro-plaintiff factors, in this case just $suffInc$ and at most all the pro-defendant factors. Her case is therefore a stronger case for the plaintiff than Hans' case, and the court must therefore decide her case in her favor.

To see an example of the constraint not applying, we consider the new fact situation of Jakob:

$$F_j := \{suffInc, prevPet, shifts\}$$

We can see that while his fact situation does contain at least all the pro-plaintiff factors, it also contains a pro-defendant factor that was not present in Hans' case. Therefore, Jakob's fact situation is not a stronger case for the plaintiff, and the result model does not force any decision for the plaintiff.
Clearly, since there is no case in the case base that was decided for the defendant, there also cannot be any precedential constraint that forces a decision for the defendant, thus the court is permitted to rule for either side in Jakob's case.

2.3 The factor-based reason model for precedential constraint

The first of the more complex models we present is the factor-based reason model for precedential constraint, specifically the version published by Horty and Bench-Capon in [HB12]. Horty actually first published his reason model in [Hor11], where he motivates the development of the reason model by arguing that the constraint of the result model is quite weak, and in particular does not match real world precedential reasoning well.

The central idea of the reason model is to focus on the specific *reason* the court provides to justify its decision, instead of relying entirely on the strength of the arguments for each side. This justification of the decision takes the form of a rule, which presents an argument, or a reason, which the court finds justifies its decision. The way that constraint is then derived from these precedents is by examining what the courts rules say about the preference of reasons. In other words, if a court gives a reason for deciding the case for the plaintiff, then it expresses that it deems this reason preferable to any reason the defendants side brought forward. Any new case must then be decided consistently with the preferences expressed by the precedent decisions. One obvious advantage of this reason based approach is that it enables courts to provide meaningful rules even in cases that are overwhelmingly strong for one side, which we will show with an example.

Example 2. Consider a case with a large amount of important factors speaking for the defendant, and very few for the plaintiff.

$$\left\langle \left\{ f_1^\delta, f_2^\delta, f_3^\delta, f_4^\delta, f_5^\delta, f_1^\pi, f_2^\pi \right\}, \delta \right\rangle$$

This case would provide almost no value in the result model, as only a case that is even stronger for the defendant will be constrained, even though that decision would have never been in doubt. If the court can specify a rule however, it can select just those factors that are sufficient for justifying the decision for the plaintiff. Suppose the court finds that the factors f_3^δ and f_5^δ alone are already a sufficient reason to rule for the defendant. It thereby expresses that even just the subset $\left\{ f_3^\delta, f_5^\delta \right\}$ is preferable to the factors for the plaintiff. This information can then be used to constrain a new fact situation that is weaker for the defendant according to the result model, but still presents the factors the court found to be sufficient, like

$$\left\{ f_1^\delta, f_3^\delta, f_5^\delta, f_1^\pi, f_2^\pi \right\}$$

As we can see, the reason model has a stronger form of constraint, because now cases that would be unconstrained by the result model are constrained by the reason model. To see how the reason model implements this idea exactly, we again start by adapting the representation of a case, before defining the constraint.

Knowledge Representation To begin, we need to change the representation of a case to include the courts argument for deciding for the chosen outcome. For this we will formally define the notions of a *reason* and a *rule*.

Definition 6 (Reason)**.** Given a set of factors $F = F^s \cup F^{\overline{s}}$, a reason is a nonempty subset $U \subseteq F$ such that U contains factors that all favor the same side. So we have either $U \subseteq F^s$ or $U \subseteq F^{\overline{s}}$. A reason U is at least as strong as another reason V if $V \subseteq U$.

We say that a reason favors outcome s or that a reason is a reason for s, if all the factors of the reason favor outcome s. In order to talk about reasons and how they relate to fact situations, we introduce the notion of *reason satisfaction*, which is defined very naturally.

Definition 7 (Reason satisfaction)**.** A fact situation F satisfies a reason U, written

$$F \vDash U$$

if and only if $U \subseteq F$.

Example 3. Using the factors from our running example, the set $\{suffInc, prevPet\}$ is a reason that favors the plaintiff. The set $\{smallApp\}$ is a reason that favors the defendant. The set $\{suffInc, prevPet, shifts\}$ is not a reason, as it contains factors for the plaintiff, but also a factor for the defendant. The fact situation $\{suffInc, smallApp\}$ satisfies the reason $\{smallApp\}$, so we have

$$\{suffInc, smallApp\} \vDash \{smallApp\}$$

however the fact situation does not satisfy the reason $\{suffInc, prevPet\}$. We write this as

$$\{suffInc, smallApp\} \nvDash \{suffInc, prevPet\}$$

The next definition is that of a rule. As we described in the beginning, the rule is provided by the court and provides a justification for the decision.

Definition 8 (Rule, factor-based model)**.** A rule $r = U \rightarrow s$ consists of a reason U and an outcome s. The outcome favored by the reason must match the outcome of the rule.

The first part of a rule r is the *premise*, and we will write $Premise(r)$ to refer to it. Similarly, we write $Outcome(r)$ to refer to a rules outcome.
Finally, we can use these definitions to modify our definition of a case.

Definition 9 (Case, factor-based reason model)**.** A case $c = \langle F, r, s\rangle$ consists of a fact situation F, a rule r and an outcome s. The premise of the rule must be satisfied by the fact situation, and the outcome of the rule must match the outcome of the case. Formally, that means

$$F \vDash Premise(r)$$

and

$$Outcome(r) = s$$

Example 4. Suppose the shelter has in the past decided the applications of Ursula and Victor. We will denote their cases by u and v. The factors in Ursula's case are $smallApp, prevPet, suffInc$, and the shelter decided in favor of the plaintiff, providing the rule

$$r_u := \{prevPet\} \rightarrow \pi$$

as justification.
Victors application also contains the factors $prevPet$ and $suffInc$, along with $shifts$. The shelter decided in favor of the defendant, using the rule

$$r_v := \{shifts\} \rightarrow \delta$$

Their cases are then

$$u = \langle \{smallApp, prevPet, suffInc\}, \{prevPet\} \to \pi, \pi \rangle$$
$$v = \langle \{prevPet, suffInc, shifts\}, \{shifts\} \to \delta, \delta \rangle$$

Having modified the representation of a case to include a rule, we can now move on to defining the precedential constraint of the reason model.

Constraint As we mentioned in the beginning, the way to obtain constraint from the rules a court specifies is through the preference of reasons that the rules express. The first question is therefore, what exact preferences can be derived from a rule. Suppose the court writes the rule r with outcome s, then the first observation is that clearly the reason $Premise(r)$ is preferred to the strongest reason for $\overline{s}$. Were this not so, the court would have used that strongest reason for $\overline{s}$ to rule for $\overline{s}$ after all. However, there are additional preferences we can derive from the rule. First, if the court prefers $Premise(r)$ to the strongest reason for $\overline{s}$ that is satisfied by the fact situation, then clearly it also prefers $Premise(r)$ to any other reason for $\overline{s}$ satisfied by the fact situation. Secondly, if the court prefers $Premise(r)$ to the strongest reason for $\overline{s}$, then it naturally prefers a reason stronger than $Premise(r)$ also.
Using these intuitive observations, we can formally define a preference relation on reasons, first for an individual case and then for a case base.

Definition 10 (Preference relation of a case)**.** Let $c = \langle F, r, s \rangle$ be a case. Let further U and V be two reasons for $\overline{s}$ and s respectively. We then have that reason V is preferred to reason U, written

$$U <_c V$$

if and only if

1. $Premise(r) \subseteq V$
2. $F \models U$

We can now lift this definition to the level of a case base.

Definition 11 (Preference relation of a case base)**.** Let Γ be a case base, and let U and V be two reasons for s and $\overline{s}$ respectively. Then we have

$$U <_\Gamma V$$

if and only if

$$U <_c V$$

for some case $c \in \Gamma$.

Example 5. We use the cases of Ursula and Victor from Example 4.

$$u = \langle \{smallApp, prevPet, suffInc\}, \{prevPet\} \rightarrow \pi, \pi \rangle$$
$$v = \langle \{prevPet, suffInc, shifts\}, \{shifts\} \rightarrow \delta, \delta \rangle$$

From the case u we can obtain the following preferences:

$$smallApp <_u prevPet$$
$$smallApp <_u prevPet, suffInc$$

We can see that although the rule of u only gives us the first pair explicitly, we can derive the second pair from it, as $prevPet \subseteq prevPet, suffInc$. From the case v we obtain

$$prevPet, suffInc <_v shifts$$
$$prevPet <_v shifts$$
$$suffInc <_v shifts$$

Here we can see that not only is $shifts$ preferable to both factors, it is obviously then also preferable to just one or none of the factors.
In the case base containing both cases, we would then have all the preferences from above.

Using this preference relation, we can move on to the way it is used to constrain future decisions. As we mentioned above, the courts are required to decide any new case consistently with the precedent cases. For the reason model, we formally define consistency, and base it on the preference relation we just defined.

Definition 12 (Inconsistent and consistent case bases)**.** Let Γ be a case base with its preference relation $<_\Gamma$, then Γ is inconsistent, if and only if there are two reasons U and V, such that $U <_\Gamma V$ and $V <_\Gamma U$, and it is consistent if and only if it is not inconsistent.

The constraint a case base imposes onto a court can now be defined using the consistency of a case base.

Definition 13 (Constraint, forcing decisions, permitting decisions)**.** Given a consistent case base Γ and a new fact situation F, the court is forced to decide F for side s, if and only if every rule r with outcome $\overline{s}$ leads to an inconsistent case base. Formally, if for every well-defined case

$$c = \langle F, r, \overline{s} \rangle$$

the case base

$$\Gamma \cup \{c\}$$

is inconsistent.
The court is permitted to decide F for side s if and only if there exists a rule with

outcome s, such that including the case resulting from it leads to a consistent case base. Formally, if there exists a well-formed case

$$c = \langle F, r, s \rangle$$

such that the case base

$$\Gamma \cup \{c\}$$

is consistent.

To see how the definition of constraint works, we will now consider some examples. However, to allow us to speak in a precise way about the options the court has when presented with a new case, we need to introduce some more terminology first, namely the concepts of *applicable rules*, and of *following* and *distinguishing* rules.

Definition 14 (Applicable rules, following a rule, distinguishing a rule)**.** Given a precedent case $c = \langle F, r, s \rangle$ and a new fact situation F_{new}, the rule r is applicable to F_{new} if the case

$$\langle F_{new}, r, s \rangle$$

is a well-formed case. In particular, that means that

$$Premise(r) \subseteq F_{new}$$

holds.
Given a case base Γ and a new fact situation F_{new}, we say that a court follows a rule, if the rule r used to decide F_{new} is the rule of a case in the case base.
We say that the court distinguishes a rule, if there is an applicable rule r in the case base, but the court chooses to make a new rule, instead of following the applicable rule.

When distinguishing a rule, the court naturally rules for the opposite outcome of some applicable rule.
With these concepts, we can now move on to some examples.

Example 6. We will use the case base we just built in Example 4, with the cases of Ursula and Victor.

$$u = \langle \{smallApp, prevPet, suffInc\}, \{prevPet\} \rightarrow \pi, \pi \rangle$$
$$v = \langle \{prevPet, suffInc, shifts\}, \{shifts\} \rightarrow \delta, \delta \rangle$$

For a reminder on the exact preference relation derived from these two cases, see Example 5.
To see how the model uses the precedent cases in the case base, we must consider some new cases. We will use the cases of William and Tara. We begin with William's application. He presents the fact situation:

$$F_w = \{shifts, smallApp, prevPet\}$$

Of the two established rules, both are applicable, as

$$premise(r_u) = prevPet \subseteq \{shifts, smallApp, prevPet\} = F_w$$

and

$$premise(r_v) = shifts \subseteq F_w$$

Is the court then permitted to rule for both outcomes? Let us first see if the court is permitted to rule for the plaintiff. For this, we need to consider all possible rules with outcome π, and as $prevPet$ is the only factor favoring the plaintiff, the rule must be identical to the established rule r_u. However, including the case

$$w = \langle F_w, r_u, \pi \rangle$$

yields an inconsistency. As now, we have $shifts <_w prevPet$, which contradicts $prevPet <_v shifts$. This means that the court is not permitted to rule for the plaintiff. Is the court then permitted to rule for the defendant? Yes, as we can extend the case by

$$w = \langle F_w, r_v, \delta \rangle$$

which is permissible, as the new case base contains no inconsistencies. The only changes to the preference relation are the inclusion of $prevPet <_\Gamma smallApp, shifts$.
Since there is only one possible rule with outcome π, and it does not lead to a consistent case base, we get the result that the court is in fact forced to decide William's case for the defendant.
Now we consider Tara's application. She presents the fact situation

$$F_t = \{shifts, smallApp, prevPet, spouse\}$$

with $spouse$ being a new pro-plaintiff factor, standing for the circumstance that the applicant's spouse is living in the same apartment. As with Williams case, we can see that both rules are applicable. Is the court permitted to decide for both outcomes this time, or is it again forced to decide for a certain outcome? Let us begin this time by checking if the court is permitted to decide for the defendant. This is indeed the case, as the shelter could follow the rule r_v again and decide in favor of the defendant. This would again simply extend the preference relation, but not introduce any inconsistency. Is the court permitted to rule for the plaintiff? In this case yes, as there is a new factor of $spouse$, and the shelter can use it to distinguish the rule. A possible decision in favor of the plaintiff could use for example the following rule:

$$r_t = \{prevPet, spouse\} \to \pi$$

The extension of Γ with $t = \langle F_t, r_t, \pi \rangle$ introduces no inconsistency and is therefore permitted.
The intuition behind the two cases is quite clear. In William's case, the precedent case of Victor already determined that working in shifts justifies a decision for the defendant,

even when the applicant has previously owned a pet and has sufficient income. William does not provide any new aspect in his favor, and therefore the court being forced to decide against him seems appropriate. In Tara's case, the decision to go against the rule established in Victor's case could be motivated by the belief, that considering that a spouse is present as well as that she previously owned a pet, the fact she works in shifts is not an issue. By distinguishing the rule, the court essentially provides an exception to the rule under specific circumstances.

Having covered a very basic precedential reasoning model in the result model, as well as the original reason model we can move on to the modifications to this model that generalize its ideas.

2.4 The reason model with inconsistent case bases

The first such modification addresses a restriction of the reason model that is of great concern for the practical application, namely that the case base needs to be consistent at all times. We can see in Definition 13 that we require the case base to be consistent to begin with. In practice it is highly unlikely that any case base will be completely free from contradictions. This could either be due to the case base containing cases from many years, during which some decisions were made without properly researching whether they contradicted some previous decision, or simply because two authorities disagreed on how to decide certain situations[3]. For this reason, Canavotto proposed a generalization that is able to derive constraint from a case base, even if that case base is inconsistent [Can22, Canng]. The intuition of the approach is, that while the case base might not be consistent to begin with, we can at least make sure it does not get more inconsistent with additional decisions. This of course requires some measure of inconsistency. To capture this, Canavotto considers the specific inconsistencies in a case base. An inconsistency is any situation which expresses a contradiction in the decisions and preferences of the court. This approach then leads to a rather simple concept of what it means for the case base to get more inconsistent, by checking whether there is a new instance of preferences contradicting. Since the knowledge representation using rules is unchanged from the reason model we just saw, we begin directly be presenting the generalized definition of constraint.

Constraint In order to define what it means for the case base to not get more inconsistent, we first define an inconsistency. Recall that consistency of a case base was defined in terms of the preference relation on reasons.

Definition 15 (Inconsistency)**.** Given a case base Γ with its preference relation $<_\Gamma$, an inconsistency of Γ is a pair of reasons U and V such that both $U <_\Gamma V$ and $V \leq_\Gamma U$. We

[3] For an example, consider the nanny robot from [CH22]. If the case base of such a robot was created by having multiple groups of parents fill out a questionnaire, it seems almost impossible that all their decisions are consistent with one another, as different parents will have different priorities.

write such an inconsistency as

$$U \perp_\Gamma V$$

We denote the set of all inconsistencies of a case base Γ by $Inc(\Gamma)$. With this definition, we have a way of expressing whether a case base gets more inconsistent if we add a new decision, simply by comparing the set $Inc(\Gamma)$ before and after the addition of the new decision. This leads to the new definition of constraint.

Definition 16. Given a case base Γ and a new fact situation F, the court is forced to decide F for side s, if and only if for every rule r with outcome $\overline{s}$ the set of all inconsistencies of Γ strictly grows when adding the new decision. Formally, if for every well-defined case

$$c = \langle F, r, \overline{s} \rangle$$

we have that

$$Inc(\Gamma \cup \{c\}) \nsubseteq Inc(\Gamma)$$

The court is permitted to decide F for side s if and only if there exists a rule with outcome s, such that including the case resulting from it does not change the set of all inconsistencies of Γ. Formally, if there exists a well-formed case

$$c = \langle F, r, s \rangle$$

such that

$$Inc(\Gamma \cup \{c\}) = Inc(\Gamma)$$

To see how this generalization affects the model, we consider an example.

Example 7. Let Γ be the case base that contains the cases of Ursula and Victor from Example 4 along with the case of William from Example 6. Instead of deciding William's case for the defendant however, as in the example, we assume that the court decided for the plaintiff, following the rule r_u from Ursula's case to justify the decision. As we saw in Example 6, this decision is inconsistent with those of Ursula and Victor, which is why it was prohibited under the constraint of the reason model. We assume however, that the decision still happened, leading to our case base being inconsistent.
This leaves us with the case base containing the cases

$$\begin{aligned} u &= \langle \{smallApp, prevPet, suffInc\}, \{prevPet\} \to \pi, \pi \rangle \\ v &= \langle \{prevPet, suffInc, shifts\}, \{shifts\} \to \delta, \delta \rangle \\ w &= \langle \{shifts, smallApp, prevPet\}, \{prevPet\} \to \pi, \pi \rangle \end{aligned}$$

First, let us recall again the specific inconsistency in our case base. When discussing William's case, we saw that using the rule

$$r_u = \{prevPet\} \to \pi$$

lead to an inconsistency, as we got both $shifts <_w prevPet$, and $prevPet <_v shifts$ from William's and Victor's case respectively. This is the only inconsistency in the case base so we have

$$Inc(\Gamma) = \{shifts \perp_\Gamma prevPet\}$$

With this case base, we now examine the new case of Zoe, with the fact situation

$$F_z = \{smallApp, prevPet, spouse, suffInc\}$$

Is the court permitted to decide for the plaintiff? Indeed it is, because using the rule r_u as justification introduces no new inconsistencies. None of the preferences expressed by this new case contradict any established preference, so we cannot introduce any new inconsistencies.
The court is not however permitted to decide for the defendant. The only possible rule for the defendant is $r_z = \{smallApp\} \to \delta$, but if we look at the preference relation of the case

$$z = \langle F_z, r_z, \delta \rangle$$

we see that it includes the preference $prevPet <_z smallApp$, which contradicts the established preference $smallApp <_u prevPet$ from Ursula's case. This means, that the case base $\Gamma \cup \{z\}$ contains the new inconsistency $prevPet \perp_\Gamma smallApp$. The court is therefore not permitted to decide Zoe's case for the defendant.

Finally, we need an example for the situation that a decision for either outcome will lead to an existing inconsistency. For this, we assume Zoe's fact situation to instead be

$$F_z = \{prevPet, shifts\}$$

In this case, clearly all possible decisions will contradict one of the precedents. If the court rules for the plaintiff using the rule r_u then it will contradict the decision of Victor's case, just like in the case of William. If the court rules for the defendant using the rule r_v then it contradicts William's case. In either case, the inconsistency that results from deciding Zoe's case is not a new one, and therefore the generalized reason model permits the court to decide for either of the two outcomes.

The most important element of this generalization is how it resolves the final situation in the example, that a new decision contradicts a precedent case no matter what outcome it has. In the original reason model, this was not a necessary consideration, as the consistency of the case base was given in the beginning and it has been shown by Horty that in that scenario there is always a possible decision that does not contradict any precedent case [HB12]. In the generalized model there may already be inconsistencies in the precedent cases. The result is that a new fact situation might contradict a case for both outcomes. A crucial observation of Canavotto is, that this is the case exactly if there is simultaneously a contradicting precedent, as well as a supporting precedent. A decision then always contradicts the one, while following the other.

The answer that the generalized model gives us in this case is that in such a scenario, both outcomes are permitted. Only if a decision causes a contradiction without following any precedent it is prohibited, as the inconsistency it would create is not yet part of the case base.
In [Canng], the author discusses this concept in more detail and points out one more option that this generalization offers. While the solution to allow both outcomes in the case that they follow one and contradict another precedent is sufficient to enable reasoning based on inconsistent case bases, it might even offer an objective way to still find some preference between the two outcomes. Such an idea might be to count how many precedent cases support and how many cases contradict a potential decision, making one outcome the preferred outcome.

The next step is to introduce additional modifications that change the case representation, to allow for more precise descriptions of the case facts.

2.5 Horty's dimension-based reason model

The next modification of the reason model, published by Horty [Hor19, Hor20], introduces a big change to the model by moving away from representing the facts of a case using factors and using *dimensions* instead. Dimensions, like factors, are used to represent the case information that have been used in legal reasoning for a while, with the difference that dimensions express a value of a certain property, instead of the mere presence or absence of some fact pattern.

The motivation to use dimensions instead of factors is clear, the factor-based representation is less detailed and has less expressive power. Since factors are binary, either present or absent, they struggle to represent facts that are more nuanced. Looking at an example from the legal context, consider the factor that the defendant damaged the property of the plaintiff. A factor-based representation would be unable to represent the scale of damage, as breaking a window and breaking a valuable artwork would both have to be reduced to a single factor representing property damage.
The solution is a more expressive representation that can express specific values for certain aspects of a case. Dimensions are used to provide this more expressive representation.

To see how this idea is implemented, we first formally define what a dimension is, and then present a modification of Horty's reason model by Horty himself [Hor19, Hor20] that uses dimensions to represent the facts of the cases. While the case representation is rather straight forward, defining constraint based on cases that use dimensions turns out to hide quite a few challenges. Horty reuses his approach of a preference relation on reasons to define consistency, which allows him to then also define constraint based on consistency of the case base. However, the definition of a reason must naturally change, since we no longer have the factors that made up the reasons before.

The way Horty defines a reason in the dimensional setting is by introducing a new sort of factor, which he calls *magnitude factor*. From now on, we will call his model the *magnitude factor model* for this reason. These new factors do not express a full fact pattern by themselves, as the regular factors do, but instead they express the fact that the value of a certain dimension is higher or lower than some reference value. The court is free to set these reference values, which means that the court creates the set of magnitude factors through that choice of reference values. The magnitude factors that can then be used to form reasons exactly like the basic factors we saw before.

To see how this process could work, we can imagine how a court might approach a case with a dimensional fact situation: One could imagine that the court looks at the fact situation and identifies some dimensions, the values of which exceed some reference value for each of the dimensions, and the court claims that this circumstance can be used to justify a decision for one outcome. This fact would be expressed in one magnitude factor for each such dimension, with the reason then containing all these magnitude factors. Other dimensions and their values might not be considered necessary to make the reason strong enough to justify the decision, and are thus left out of the reason. Once the court has found a reason it finds to justify a decision, it creates a rule with this reason and decides the case. The rest of the dimension-based reason model then follows the factor-based version, with the consistency of a case base defined on the preference of reasons expressed by each decision.

To see how the magnitude factor model works exactly, let us begin with the modification of the case representation before defining the modified precedential constraint.

Knowledge Representation We need to first define dimensions, and then redefine fact situations to now be based on dimensions instead of factors. Important to note is, that while factors always favor one or the other outcome, a dimensions values only express support for a side in relation to other values.

Definition 17 (Dimension)**.** A dimension d is a set of values that are totally ordered by a relation $\preceq_d^s$. For the relation, the properties of a partial order must hold:

- $p \preceq_d^s p$
- if $p \preceq_d^s q$ and $q \preceq_d^s t$ then $p \preceq_d^s t$
- if $p \preceq_d^s q$ and $q \preceq_d^s p$ then $p = q$

Aside from this relation, we define $\preceq_d^{\bar{s}}$ to be the dual relation, with

$$p \preceq_d^s q \text{ if and only if } q \preceq_d^{\bar{s}} p$$

We will name dimensions similarly to factors before, with $d_1, d_2, \ldots$ for generic dimensions, and telling names in examples.

Example 8. The factors used in our shelter application can easily be transformed into dimensions. Instead of the factor $suffInc$ for example, we now use the dimension $income$, with values in the interval $[0, \infty)$. We can similarly convert the other factors into dimensions as follows:

$$\begin{array}{cccc} smallApp & \rightarrow & sqm & : & [0,\infty) \\ prevPet & \rightarrow & pet & : & \{none, fish, cat, dog\} \\ shifts & \rightarrow & work & : & \{shifts, regular, home\} \\ spouse & \rightarrow & roommate & : & \{no, yes\} \end{array}$$

For the dimension pet we use the four options that the applicant never owned a pet, that the applicant owned a fish in the past, that the applicant owned a cat in the past, or that the applicant already owned a dog in the past. The dimension $work$ contains values for the applicant working in shifts, working a regular schedule, i.e. morning to afternoon, and working from home. Finally, the dimension $roommate$ contains the two values no and yes, for the applicant having no roommate, or having a roommate.
We can see from this example that the values can be numerical or any other set, as long as we provide an ordering that expresses which value favors which side over another value. We can even use a set with two values, which somewhat resembles a factor[4].
For our dimensions, this ordering should be quite intuitive. For both $income$ and sqm, higher numbers favor the plaintiff, while lower numbers favor the defendant. Formally that means that for the dimension $income$, we have

$$x \preceq^{\pi}_{income} y :\Leftrightarrow x \leq y$$

and similar for sqm. For the nonnumerical dimensions, we need to specify the order explicitly. For pet we get

$$none \preceq^{\pi}_{pet} fish \preceq^{\pi}_{pet} cat \preceq^{\pi}_{pet} dog$$

and for $work$

$$shifts \preceq^{\pi}_{work} regular \preceq^{\pi}_{work} home$$

while for $roommate$ we simply have

$$no \preceq^{\pi}_{roommate} yes$$

When defining a fact situation based on dimensions, some care needs to be taken. The approaches we present assume that for each setting that the model is used in, there is a fixed set of dimensions, and each case assigns *every* dimension a value [Hor19, Pra21]. This is of course a limitation, however as with factors, we do not explore the subject of absence of dimensions from cases, and instead simply adopt the viewpoint of the authors before us.

[4] The question of a mixed model containing both factors and dimensions is more complicated than just using a dimension with two values. However, as it is not the focus of this work, we do not discuss the issue or proposed solutions further. For additional information, see Section 6 of [Pra21].

Definition 18 (Domain, dimensional fact situation)**.** A domain is a set D of dimensions d that apply to a certain kind of case. A dimensional fact situation F is a set of value assignments of all dimensions of a domain,

$$F = \{(d, p) \mid d \in D\}$$

with the requirement that p is a value in the dimension d.

Given a fact situation F we will write $F(d)$ to denote the value of the dimension d in the fact situation F.

We will refer to dimensional fact situations as simply fact situations when it is clear from context whether we are speaking of the factor-based, or the dimension-based kind.

Example 9. To show how the representation of facts using dimension works, we will revisit the cases of Ursula and Victor from Example 4.

We will restrict our domain to the four dimensions we obtained for the original factors, using the dimensions

$$sqm, pet, income, work$$

in Example 8. The fact situations of Ursula and Victor will then contain additional information compared to the factor-based fact situations.

Beginning with Ursula, her factor-based fact situation was

$$smallApp, prevPet, suffInc$$

which are now dimensions, and thus need a value. There is also the dimension of *work* which did not play a role in the factor-based representation, but which still has to be assigned a value. Suppose then that her apartment is 80 square meters large, she previously owned a cat, her income is 1000 and she has a regular work schedule. Her fact situation is then

$$F_u = \{(sqm, 80), (pet, cat), (income, 1000), (work, regular)\}$$

Victor's apartment has 110 square meters, his income is 900, he did not own a pet in the past, and he works in shifts. His fact situation is then

$$F_v = \{(sqm, 110), (pet, none), (income, 900), (work, shifts)\}$$

With this definition of a fact situation in place, we can now turn to the question of how to define reasons. As we mentioned above, Horty defines so called *magnitude factors* that are based on the values of dimensions, which allows the rest of the model to follow the factor-based scenario. However we will see in the following Section 2.6 that there are also other approaches, which end up influencing the constraint we obtain from the model.

Recall that the idea of a magnitude factor is to express the fact that a certain dimension is higher or lower (in other words, more or less favorable for an outcome) than some reference value. This is expressed in the following definition.

Definition 19 (Magnitude factors)**.** Given a dimension d, a magnitude factor for the side s is a factor

$$M^s_{d,p}$$

with p being the *reference value* of the magnitude factor. A fact situation F satisfies a magnitude factor $M^s_{d,p}$ if and only if the value of d in F is at least as strong for s as p. That means formally, that

$$p \preceq^s_d F(d)$$

and we write it as before.

$$F \vDash M^s_{d,p}$$

To better understand this concept and how magnitude factors are used, we will look at an example.

Example 10. Let us again use the cases of Ursula and Victor with their dimensional fact situations from Example 9. That means we have the fact situation of Ursula

$$F_u = \{(sqm, 80), (pet, cat), (income, 1000), (work, regular)\}$$

and that of Victor

$$F_v = \{(sqm, 110), (pet, none), (income, 900), (work, shifts)\}$$

A court might find, that if a plaintiff has previous pet ownership experience of a cat or better, that this circumstance favors the plaintiff. The magnitude factor for this situation would then be

$$M^{\pi}_{pet,cat}$$

and since Ursula has previously owned a cat, her fact situation F_u would satisfy the magnitude factor. Formally speaking, we have

$$F_u \vDash M^{\pi}_{pet,cat}$$

because

$$cat \preceq^{\pi}_{pet} cat = F_u(pet)$$

Similarly, the court might find that an income less or equal than 950 is a factor that speaks for the defendant. This factor would be

$$M^{\delta}_{income,950}$$

and Victor's fact situation F_v satisfies it, because

$$950 \preceq^{\delta}_{income} F_v(income)$$

while for Ursula's fact situation we have

$$F_u(income) = 1000 \preceq^{\delta}_{income} 950$$

so it does not satisfy the factor.

We can see that this type of factor is indeed a factor like the ones we saw before, as in any fact situation it is either satisfied or not satisfied, and if it is satisfied, it always favors the same side. Therefore we can treat it like the standard factors, and define a reason for the dimensional case exactly as we did in the standard case.

Definition 20 (Reason, reason satisfaction)**.** A reason for side s is a set of magnitude factors for side s.
A fact situation F satisfies a reason U if and only if F satisfies every factor in U.

With the definition of a reason that consists of factors, we can now define a rule, a case and a case base as in the factor-based case.

Definition 21 (Rule, magnitude factor model)**.** A rule $r = U \to s$ consists of a reason U and an outcome s. The outcome favored by the reason must match the outcome of the rule.

Definition 22 (Case, magnitude factor model)**.** A case $c = \langle F, r, s \rangle$ consists of a fact situation F, a rule r and an outcome s. The premise of the rule must be satisfied by the fact situation, and the outcome of the rule must match the outcome of the case. Formally, that means

$$F \vDash \mathit{Premise}(r)$$

and

$$\mathit{Outcome}(r) = s$$

Definition 23 (Case base, magnitude factor model)**.** A case base is a set Γ of cases.

Example 11. Let us represent the decisions of the court for the dimensional fact situations of Ursula and Victor from Example 9.
Suppose that the court found that having an income greater or equal to 950 was a factor favoring the plaintiff, so the court formulated the magnitude factor

$$M^{\pi}_{\mathit{income},950}$$

Using this factor, the court then provides the rule

$$\left\{M^{\pi}_{\mathit{income},950}\right\} \to \pi$$

Ursula's case is then

$$u = \left\langle F_u, \left\{M^{\pi}_{\mathit{income},950}\right\} \to \pi, \pi \right\rangle$$

Similarly, the court found that having a work schedule that is shift work or anything more favorable for the defendant is a factor for the defendant, as well as having pet ownership experience that is at the level of fish or below is a factor for the defendant. These two findings lead to formulating the magnitude factors

$$M^{\delta}_{\mathit{work},\mathit{shifts}} \text{ and } M^{\delta}_{\mathit{pet},\mathit{fish}}$$

and the court uses them to provide the rule

$$\left\{M^{\delta}_{work,shifts}, M^{\delta}_{pet,fish}\right\} \to \delta$$

to decide Victor's case in favor of the defendant. His case is then

$$v = \left\langle F_v, \left\{M^{\delta}_{work,shifts}, M^{\delta}_{pet,fish}\right\} \to \delta, \delta\right\rangle$$

And we again get a case base containing both cases.

Constraint The next step would be to define the preference relation. The intuition of the preferences expressed by a decision still holds, meaning that the premise of the rule, as well as any stronger reason for the same side are preferred to any reason for the opposite side. However, the concept of the strength of a reason for a side needs a modification. While the intuition that if a reason U is a subset of another reason V then certainly V is at least as strong as U still holds, it fails to capture all situations where intuition would tell us that one reason is stronger than another. We will use an example to show this.

Example 12. Consider the reasons

$$U = \left\{M^{\pi}_{sqm,80}\right\}$$

meaning that the court should decide for the plaintiff, because their apartment is larger than 80 square meters, and

$$V = \left\{M^{\pi}_{sqm,120}\right\}$$

meaning that the court should decide for the plaintiff, because their apartment is larger than 120 square meters. Clearly, no reason is an actual subset of the other since the two magnitude factors use different reference values and are therefore distinct, but our intuition tells us, that V being satisfied by some scenario is a stronger argument for the plaintiff than only U being satisfied. We can also observe, that if V is satisfied by some scenario, then U must also be satisfied, as any apartment larger than 120 square meters must clearly also be larger than 80 square meters.

It is the final observation that leads us to the definition of *reason entailment*, which will be the new basis of determining whether a reason is stronger for a side than another reason.

Definition 24 (Reason entailment)**.** Let U and V be two reasons for side s. Then U entails V, written

$$U \Vdash V$$

if and only if it holds that for every fact situation F, if F satisfies U then F also satisfies V.

We say then, that if a reason U entails a reason V, that U is at least as strong for its side as V. Using this notion of strength for a side, we can now express the condition of the preference relation on reasons for the dimensional setting. To recall, the two conclusions we drew from a courts justification for the decision for outcome s, which is represented in the premise of the rule r, were that the court prefers a reason U to a reason V if

i. U is a reason for s and is at least as strong for s as $Premise(r)$

ii. V is any reason for $\overline{s}$ that is satisfied in the fact situation of the case

We can formalize these two conditions, using the notions of reason satisfaction from Definition 20 and of relative strength of a reason through reason entailment from Definition 24. The formal conditions are then that from the decision of a case c we derive that the court prefers reason U to reason V if

1 $Premise(r) \Vdash V$ and

2a $F \vDash U$

where F is the fact situation and r the rule of the case c.
Using exactly these two conditions will expose another issue with the dimensional reasons. In his 2019 paper, Horty is already aware of this problem, which causes his reason model to collapse into a dimensional result model[5]. We will provide the basic intuition of the flaw, before modifying the conditions for preference, and defining the preference relation following Horty's modification from [Hor20].

Example 13. Suppose we have a case base containing the case of of Tim, with a domain restricted only to *income*, for simplicity. Let Tim's case be the following:

$$t = \left\langle (income, 1000)\,, \left\{ M^{\pi}_{income,800} \right\} \rightarrow \pi, \pi \right\rangle$$

which means, that the court decided in favor of Tim, with the reason that his income was greater or equal to 800.
Let us now consider a new fact situation, brought by Kirsten:

$$F_k = (income, 950)$$

When considering her fact situation and the rule the court provided in Tim's case, our intuition should tell us that as Kirsten's income surpasses the threshold specified in the rule, the decision should also be in her favor. In particular, a decision against Kirsten would feel contradictory to the rule established by Tim's case. Of course, one could point to Kirsten's income being less favorable for her side than Tim's was for his. For a fortiori reasoning this would suffice as a justification for deciding against Kirsten. However, the

[5] For a detailed discussion of the issue and a formal proof see [Hor19]

idea of the reason model is exactly to allow for the court's reason to allow for more detailed reasoning. And the court's rule would tell us to rule in favor of Kirsten.
Let us then examine what happens when the court decides against Kirsten, providing the rule

$$\left\{M^{\delta}_{income,951}\right\} \to \delta$$

meaning that because her income is not greater than 951, a decision against her is justified. The two rules clearly are problematic, since the one rule says an income over 800 is reason to decide for the plaintiff, while the other says an income under 951 is reason to decide for the defendant.
However, the second rule would be permitted in the dimensional reason model, if we define the preferences expressed by the court using only the two conditions we described above. The problem lies in the fact, that the premise of Kirsten's rule is not satisfied by Tim's fact situation, so condition 2a is not met. That means, that according to Tim's case, the court does not prefer his rule's premise to that of Kirsten's rule.

The solution for this problem that Horty provides in [Hor20] is quite simple. Basically, we need a way to add to our conditions of preference that if the court uses a magnitude factor with some reference value to justify a decision for one side, that this establishes a preference over all magnitude factors for the same dimension and reference value that favor the other side. In simple terms, if the court uses the magnitude factor $M^{s}_{d,p}$ then this establishes that dimension d having a value more favorable to side s than reference value p is a preferred reason to d having a value more favorable to side $\overline{s}$ than p.
To capture this idea formally, we first introduce a piece of notation, which is then used to modify the conditions of preference of reasons. For a reason U for side s, let

$$\overline{U} = \left\{M^{\overline{s}}_{d,p} \mid M^{s}_{d,p} \in U\right\}$$

be the reason for side $\overline{s}$ containing the inverse of all magnitude factors in U. That means, that if U contains a magnitude factor $M^{s}_{d,p}$, which favors side s given the value of dimension d is at least as favorable to s than value p, the reason $\overline{U}$ contains the factor $M^{\overline{s}}_{d,p}$ which favors side $\overline{s}$ given that the value of d is at most as favorable to side s than value p.
Intuitively, if a court includes a magnitude factor $M^{s}_{d,p}$ in its reason, it would be natural to conclude that the court prefers it to the magnitude factor $M^{\overline{s}}_{d,p}$. On the level of reasons, this gives us that if a court uses the rule r, then it prefers $Premise(r)$ to $\overline{Premise(r)}$. In the words of Example 13, by using the rule

$$\left\{M^{\pi}_{income,800}\right\}$$

the court expresses that it thinks the income being higher than 800 is a better reason for the plaintiff than the income being lower than 800 is for the defendant.
And as before, if U is a reason for $\overline{s}$ and $\overline{Premise(r)}$ is at least as strong as U, then the court would also prefer $Premise(r)$ over U. We can then observe, that $\overline{Premise(r)}$ is at

least as strong for $\overline{s}$ as the premise of Kirsten's rule, which gives us the desired effect that the court prefers the premise of Tim's rule to that of Kirsten's rule.
This now allows us to finally define the preference relation for the dimensional scenario formally.

Definition 25 (Preference relation of a case, magnitude factor model)**.** Let $c = \langle F, r, s\rangle$ be a case. Let further U and V be two reasons for $\overline{s}$ and s respectively. We then have

$$U \leq_c V$$

if and only if

1 $Premise(r) \Vdash V$ and either

2a $F \vDash U$ or

2b $\overline{Premise(r)} \Vdash U$

This new definition allows us to simply reuse the definition from the factor-based model.

Definition 26 (Preference relation of a case base, magnitude factor model)**.** Let Γ be a case base, and let U and V be two reasons for s and $\overline{s}$ respectively. Then we have

$$U <_\Gamma V$$

if and only if

$$U <_c V$$

for some case $c \in \Gamma$.

Now we provide some examples to show the concepts we just defined in some more detail.

Example 14. In the dimensional case, it is not feasible to list the entire preference relation, as there are infinitely many magnitude factors that can be combined into infinitely many reasons. We will however give examples on how to test whether two reasons are part of the preference relation or not.
The first example shows how condition 2a works, and is another example of reason satisfaction. The second example shows condition 1, and is an example for reason entailment. The third example shows condition 2b, and is an example for the $\overline{Premise(r)}$ notation. For all these examples we will use the case of Victor from Example 11. To recall, the fact situation of that case was

$$F_v = \{(sqm, 110), (pet, none), (income, 900), (work, shifts)\}$$

and the rule used in that case was

$$r_v = \left\{M^\delta_{work,shifts}, M^\delta_{pet,fish}\right\} \rightarrow \delta$$

1. We will first show that from the case v we get the preference

$$\left\{M^{\pi}_{sqm,100}, M^{\pi}_{income,800}\right\} <_v Premise(r_v)$$

Checking condition 1 is trivial, as it requires we show

$$Premise(r_v) \Vdash Premise(r_v)$$

which holds. Checking condition 2a means checking whether the fact situation F_v satisfies the reason $\left\{M^{\pi}_{sqm,100}, M^{\pi}_{income,800}\right\}$. For the two relevant dimensions, the fact situation contained the assignments

$$(sqm, 110) \text{ and } (income, 900)$$

which means, that we have

$$100 \preceq^{\pi}_{sqm} F_v(sqm)$$

and

$$800 \preceq^{\pi}_{income} F_v(income)$$

so we get that

$$F_v \vDash \left\{M^{\pi}_{sqm,100}, M^{\pi}_{income,800}\right\}$$

This means that condition 2a is met, and we do indeed get that

$$\left\{M^{\pi}_{sqm,100}, M^{\pi}_{income,800}\right\} <_v Premise(r_v)$$

2. We will now show that from the case v we get the preference

$$\left\{M^{\pi}_{sqm,100}, M^{\pi}_{income,800}\right\} <_v \left\{M^{\delta}_{work,shifts}, M^{\delta}_{pet,none}, M^{\delta}_{income,600}\right\}$$

Since we already checked that condition 2a holds, we will only show that condition 1 holds. For this, we need to show that the reason

$$U = \left\{M^{\delta}_{work,shifts}, M^{\delta}_{pet,none}, M^{\delta}_{income,600}\right\}$$

entails the premise of the rule r_v.
Intuitively it is easy to see that U is a stronger reason for the defendant than $Premise(r_v)$, as not only does it have reference values that are more favorable for the defendant, it also includes an additional factor that favors the defendant.
For the formal argument, let F be any fact situation that satisfies U. We must therefore have that

$$shifts \preceq^{\delta}_{work} F(work), none \preceq^{\delta}_{pet} F(pet) \text{ and } 600 \preceq^{\delta}_{income} F(income)$$

all hold. This tells us immediately, that F also satisfies $Premise(r_v)$ as in the premise, we only have reference values for $work$ and pet, and for those values we have

$$shifts \preceq^{\delta}_{work} F(work) \text{ and } fish \preceq^{\delta}_{pet} none \preceq^{\delta}_{pet} F(pet)$$

This means, that $Premise(r_v)$ is indeed entailed by U, and condition 1 holds, and together with condition 2a this means that the preference

$$\left\{M^{\pi}_{sqm,100}, M^{\pi}_{income,800}\right\} <_v \left\{M^{\delta}_{work,shifts}, M^{\delta}_{pet,none}, M^{\delta}_{income,600}\right\}$$

does hold.

3. For the final example using the case of Victor, we show that we get the preference

$$\left\{M^{\pi}_{pet,fish}\right\} <_v Premise(r_v)$$

Condition 1 is here again trivial. We can also see that condition 2a does not hold here quite easily, as for $\left\{M^{\pi}_{pet,fish}\right\}$ to be satisfied, the fact situation would need to contain a value for *pet* that is at least as favorable as $fish$ for the plaintiff. In Victor's case however, the value for *pet* is *none*, which is less favorable for the plaintiff.
So we need to use condition 2b to show the preference. That means showing that

$$\overline{Premise(r_v)} \Vdash \left\{M^{\pi}_{pet,fish}\right\}$$

holds. First, what is $\overline{Premise(r_v)}$? It is a reason for the opposite side than r_v, so a reason for the plaintiff, and it contains a magnitude factor for each factor in the original premise, but with the opposite outcome. So as $Premise(r_v)$ contains the factor $M^{\delta}_{work,shifts}$, $\overline{Premise(r_v)}$ contains the factor $M^{\pi}_{work,shifts}$.
Also, since $Premise(r_v)$ contains the factor $M^{\delta}_{pet,fish}$, we get that $\overline{Premise(r_v)}$ contains the factor $M^{\pi}_{pet,fish}$, which allows us then to immediately answer the question of entailment. We clearly have

$$\left\{M^{\pi}_{work,shifts}, M^{\pi}_{pet,fish}\right\} \Vdash M^{\pi}_{pet,fish}$$

as any fact situation satisfying the two factors of $\overline{Premise(r_v)}$ trivially satisfies just one of the two factors also.
This means, that the preference

$$\left\{M^{\pi}_{pet,fish}\right\} <_v Premise(r_v)$$

does hold.

We will also argue shortly that including condition 2b indeed fixes the situation presented in Example 13. The Problem in that situation was, like in the third example we just discussed, that the reason used as the premise of Kirsten's rule is not satisfied by Tim's fact situation, so condition 2a does not hold. Using condition 2b however gives us that $\overline{Premise(r_t)}$ entails $Premise(r_k)$ and we do get the preference required to make the two cases inconsistent. That the entailment holds can be checked just like we checked the entailment in the second example we discussed above.

With this working definition for the preference relation, the rest of the model follows Horty's factor-based reason model.

Definition 27 (Inconsistent and consistent case bases, magnitude factor model)**.** Let Γ be a case base with its preference relation $<_\Gamma$, then Γ is inconsistent, if and only if there are two reasons U and V, such that $U <_\Gamma V$ and $V <_\Gamma U$, and it is consistent if and only if it is not inconsistent.

Definition 28 (Constraint, magnitude factor model)**.** Given a consistent case base Γ and a new fact situation F, the court is forced to decide F for side s, if and only if every rule r with outcome $\overline{s}$ leads to an inconsistent case base. Formally, if for every well-defined case

$$c = \langle F, r, \overline{s} \rangle$$

the case base

$$\Gamma \cup \{c\}$$

is inconsistent.
The court is permitted to decide F for side s if and only if there exists a rule with outcome s, such that including the case resulting from it leads to a consistent case base. Formally, if there exists a well-defined case

$$c = \langle F, r, s \rangle$$

such that the case base

$$\Gamma \cup \{c\}$$

is consistent.

As the definition of constraint does not change from the factor-based reason model to the dimension-based reason model, we do not provide any example of a case base constraining the courts decisions. The process is the same as in Example 6, it is only checking the preference of reasons that changes, but we provided an example for this already.

2.6 An alternative dimension-based reason model

Aside from Horty's dimension-based reason model, we also want to present an alternative approach to the dimensional case. This other dimension-based reason model was published by Prakken [Pra21]. It uses the same concept of dimensions, but takes a more direct approach to the rule of a case, dropping the need for magnitude factors. The motivation for this different approach is that while Prakken claims it is less expressive, he argues it would be easier to apply in practice.
In Prakken's model he uses the same basic intuition of the court identifying certain reference values for each dimension, and then using the fact that the given case is more favorable to one side than these reference values to justify the decision. The central difference is that instead of using magnitude factors to express the fact that the value

of some dimension is more favorable than some reference value, Prakken just states the reference value directly. On top of that, Prakken requires that the court provides a reference value for *every* dimension, while in Horty's model some dimensions could be left out of the rule. We will therefore call Prakken's model the *complete rule model.* This of course means that the rules of the two approaches look very different, even though they express very similar concepts.
As Prakken's rules do not contain factors, his definition of constraint cannot rely on reasons the same way that Horty's does. Instead his definition of constraint simply compares the values of new fact situations directly against the rules of the precedent cases.

To see how exactly these definitions look, we again begin with the case representation.

Knowledge representation The basic concepts of representing fact situations using dimensions is identical to that of Horty's model. We also assume a domain consisting of a set of dimensions, and a fact situation is again an assignment that assigns each of the dimensions a value. The difference is in the rule that the court provides to justify its decision. A rule in Prakken's model provides reference values for each dimension, instead of allowing for the selection of specific dimensions. The intuition of a rule however remains close to that of Horty. The court justifies its decision for outcome s by providing a set of values that are less favorable for side s than the actual values in the case.

Definition 29 (Rule)**.** A rule

$$r = \{(d, p) \mid d \in D\} \to s$$

consists of a premise, which is a set of value assignments for all dimensions and an outcome $s \in \{\pi, \delta\}\}$.

This definition then leads directly to the definition of a case, where now the idea of the rule being a lower bound for the values is reflected in a new definition of how a rule must match the fact situation.

Definition 30 (Case, complete rule model)**.** A case $c = \langle F, r, s\rangle$ consists of a fact situation F, a rule r and an outcome s. The outcome of r must match the outcome of the case, and each dimension in $Premise(r)$ must be assigned a value less favorable to s than F. Formally, that means

$$Outcome(r) = s$$

and

$$r(d) \preceq_d^s F(d)$$

for all dimensions $d \in D$.

We will say that a rule that matches a fact situation in the way we just defined *applies* in that fact situation.
The definition of a case base naturally remains.

With this relatively simple definition of rules, we can now directly move on to the way the model constrains decisions.

Constraint The definition of constraint is relatively easy as it does not involve any additional concepts beyond the rules we just defined. Using the ordering of the values of each dimension, we can simply look for rules in the case base that apply to a new fact situation, and use them to force the decision of the court. The resulting model is therefore not too different from a dimension based result model, with the difference being that the court can set the preference values how it sees fit, which offers additional insight compared to a fortiori constraint.

Definition 31 (Constraint, complete rule model)**.** Given a case base Γ and a new fact situation F_{new}, the court is forced to decide F_{new} for side s, if and only if there exists a case $c = \langle F, r, s\rangle \in \Gamma$ such that the rule r applies to F_{new}, so that

$$r(d) \preceq_d^s F_{new}(d)$$

for all dimensions $d \in D$.
The court is permitted to decide F_{new} for side s if and only if it is not forced to decide F_{new} for $\overline{s}$.

There is an additional precaution we have to consider for the complete rule model, which arises from the fact that we use $\preceq_d^s$ in our definition, which is a "less or equally strong for side s" relation. It is therefore possible for the courts to create two rules using the same reference values but with opposite outcomes, which leads to the possibility that a court might be forced to decide a future case for both outcomes. Fixing this issue is quite straightforward, we could for example simply require that the court cannot create a rule with the same reference values as an existing rule, but for a different outcome.
We will now present an example of the complete rule model.

Example 15. Since the complete rule model uses a quite different mechanism than the magnitude factor model, we will introduce new cases for this example, but we will still use the restricted domain that only contains the dimensions

$$sqm, pet, income, work$$

These new cases are those of Maria and Linus. We will directly provide their fact situations in the formal representation, but will explain the rules in a little more detail.
The fact situation of Maria is

$$F_m = \{(sqm, 60), (pet, cat), (income, 800), (work, regular)\}$$

and for Linus we have

$$F_l = \{(sqm, 90), (pet, dog), (income, 1200), (work, home)\}$$

as the fact situation.
As for the rules, suppose that in Maria's case, the court decided in favor of the defendant, and provided the following rule:

$$r_m = \{(sqm, 80), (pet, cat), (income, 1000), (work, regular)\} \rightarrow \delta$$

This rule justifies the decision for the defendant by arguing that an applicant with an apartment at most 80 square meters large, with pet experience of at most a cat, with an income less or equal to 1000 and a work schedule that is regular or shift work should be denied. Noteworthy is here, that the value of *cat* in the dimension *pet* is quite favorable for the plaintiff, yet it is still part of a rule justifying a decision for the defendant. The intuition is here, that those dimensions with reference values that might be considered strong for the defendant can compensate for those dimensions with reference values considered weak for the defendant. In other words, one might formulate this particular case as: If the applicant does not have an apartment larger than 80 square meters and an income of more than 1000, then even pet experience up to previous ownership of a cat and a regular work schedule do not suffice to decide for the plaintiff.
In the actual rule, we do not split the fact situation in this way, but it allows for a more clear interpretation of how the courts rule provides a justification for the decision.
Now let us consider the decision of the second case, that of Linus. Suppose the court decided for the plaintiff, and provided the following rule:

$$r_l = \{(sqm, 90), (pet, cat), (income, 1000), (work, home)\} \rightarrow \pi$$

Let us now introduce new fact situations, to see how to apply the constraint of the model. Consider the application of Oscar, who presents the following fact situation:

$$F_o = \{(sqm, 80), (pet, cat), (income, 900), (work, shifts)\}$$

When comparing the fact situation to rule r_m, we can see that Oscar's fact situation is less favorable for him in each dimension. The court is therefore forced to decide his case for the defendant. This also shows that the model has stronger reasoning than a fortiori reasoning, as Oscar's case is not strictly stronger for the defendant than Maria's case, as his apartment is 80 square meters large compared to Maria's 60 square meters.
How about the fact situation of Nadine, which contains

$$F_n = \{(sqm, 100), (pet, fish), (income, 1100), (work, shifts)\}$$

When comparing it to both rules, we can see that in each case, there is a dimension which is less favorable for the outcome than the value in the rule. The court is therefore permitted to rule for either outcome.

We can see that checking constraint in the complete rule model is significantly simpler than in the magnitude factor model. We will present a formal comparison of the two accounts in Section 3.1, and continue now with presenting one more model. This model adds a hierarchical structure to the fact representation, which allows us to make the outcome of our model many-valued instead of binary.

2.7 A hierarchical dimension-based result model

This final model we will present is an adaptation from the dimension-based result model that was published by Woerkom et al. [VGPV23]. Aside from the result model, there was also a factor-based reason model variation published by Prakken et al. [vWGPV23]. The interesting aspect of these models that we want to show is placing the factors or dimensions in a hierarchy, instead of every factor or dimension being of the same level. For simplicity, and because difference between result and reason model is not the focus of this section, we will only look at the dimensional result model going forward.

Creating a hierarchy of the fact representation is motivated by a few considerations. Some of these are mostly rooted in legal concepts, which is why we will not explore them here. The main argument for using a hierarchical model that we care about comes from the issue of determining the values of dimensions. While for some dimensions it will be very easy to determine their value in a given case, usually because they represent measurable facts, other dimensions might be more abstract and rely on some interpretation of facts to find a value in the first place. An example from the context of the nanny robot we mentioned in Chapter 1 might be the age and behavior of a child. Determining the age of the child is clearly unproblematic, but assigning the behavior of the child a value relies on subjective interpretation of the child's actions. In particular, the behavior of a child might be judged based on a number of more basic criteria that are usually more measurable, such as their use of bad language, whether they cause trouble with their siblings or whether they listen to their parents.
The proposed solution is therefore quite natural. We can arrange the dimensions in a hierarchy, placing the easily measurable dimensions at the bottom as base level dimensions. The dimensions that build on these base level dimensions are then called abstract dimensions. Determining their values now relies on the values of all dimensions that are specified to have an influence on them. In the model we present, each of these intermediate determinations are essentially precedent decisions that influence future cases.
Importantly, the outcome is also represented by a dimension. This leads to the outcome being not just a binary choice of plaintiff or defendant, but instead a value on a potentially infinite scale[6].

[6] In their paper [VGPV23], Woerkom et al. use the example of a court trying to set a bail for some defendant. In this case, the value of the outcome dimension is simply the bail amount. In Section 3.2 we will explain the meaning of a dimensional outcome further and present an alternative to a classic value-based dimension.

Building an entire structure of dimensions and reasoning with it, while clearly a worthwhile consideration, is however not in the scope of this work. Adapting the model we present in this work to include a hierarchy could be part of future work, but for now we are only interested in using the idea of the outcome being a dimension to make the models output more nuanced, as mentioned in the introduction. Using a dimension gives us the many-valued outcome that would then allow the model to differentiate for example between the better of two good solutions.
Therefore, we will present an adaptation of the hierarchical model's reasoning instead of the full model. This adaptation will consider only hierarchies with two levels, so it mirrors our usual structure in that the reasoning does only a single step from basic dimensions to the outcome. However, this outcome is now a dimension, and how this is dealt with is carried over from the model in [VGPV23]. Before presenting the intuition of what we will call the hierarchical model we recall that since it is a result model, it only implements a fortiori reasoning, which means the cases do not contain a rule.

The intuition of the hierarchical model is to use precedential reasoning to impose lower or upper bounds on the outcome value of a new case. This can be seen as a generalization of the binary outcome, as we could interpret a rule with the outcome "plaintiff" to impose a lower bound on the court to decide a constrained case at least as much for the plaintiff as the precedent case. As there are only two options, this results in the constraint forcing a decision for the plaintiff. With an outcome value however, the court may cite a precedent case where this outcome value was x, and then precedential constraint forces the court to decide on an outcome value for the new case that is at least x. The mechanism when constraint applies in this case is a fortiori reasoning, which means that the only change from a basic dimension-based result model to this adaptation is the way the outcome is specified.

As always, we begin with the representation of the cases, before moving to the definition of precedential constraint.

Knowledge representation The model uses the same representation of a fact situation as Prakken's complete rule model we presented in Section 2.6, so a fact situation is an assignment of values to all dimensions of a domain. Along with the fact situation, a case then only contains an outcome, which is the most important difference to the other models we discussed so far. The outcome, instead of being either π or δ, is now an assignment of a value to an outcome dimension which we will call ω. This outcome dimension itself is naturally also ordered. The meaning of the values of the outcome dimension is left open by the authors, but at least for court decisions that result in measurable values like bail or prison time the interpretation of the outcome value is obvious. The orderinsg of the base dimensions now express that some value favors higher or lower values for ω than other values. If we consider the dimension of *income* for example, a higher value previously favored the plaintiff. With a dimensional outcome it now favors a higher value for ω instead. This allows us to simplify the notation, by

simply writing $p \preceq_d q$ to indicate that q favors higher values for ω than p.
For continuity, we may still say that higher values of ω favor the plaintiff, and lower values of ω favor the defendant, if there is no intended meaning for the outcome in the situation we are discussing. This leads to the following definition of a case.

Definition 32 (Case, hierarchical model)**.** A case $c = \langle F, (\omega, v)\rangle$ consists of a fact situation F and an assignment of a value v to the outcome dimension ω[7].

A case base is still defined as before.
Following is the definition of precedential constraint, which introduces the main feature we are interested in, constraining a many-valued outcome.

Constraint As a result model, the constraint uses a fortiori reasoning, which we have discussed in Section 2.1. The difference in this case is that the constraint of a precedent case is not to force a decision for the same outcome, but instead the precedent case provides a lower or upper bound on the value of the outcome dimension.

Definition 33 (Constraint, hierarchical model)**.** Given a case base Γ and a new fact situation F_{new}, the value v is a lower bound (or upper bound) for the courts decision of F_{new}, if and only if there exists a case $c = \langle F, (\omega, v)\rangle \in \Gamma$ such that for all dimensions, the value in F_{new} favors higher (or lower) values for ω than in F. Formally, that means that v is a lower bound for ω if and only if

$$F(d) \preceq_d F_{new}(d)$$

for all dimensions, and v is an upper bound for ω if and only if

$$F_{new}(d) \preceq_d F(d)$$

for all dimensions. The court must choose a value for ω that is greater than the greatest lower bound, and less than the least upper bound.
The court is not bound in its decision of F_{new} if and only if there is neither a lower nor an upper bound.

To see how this model works, we will consider an example.

Example 16. To show how the hierarchical model works, we need to again change our scenario slightly, to reflect the outcome now being a dimension. Previously, we said that a decision for the plaintiff was a decision to accept an application, while a decision for the defendant was a decision to deny an application. Now we consider a changed scenario, where the shelter has so many applications that it implements a waiting list. This waiting list however is not a simple first in first out queue, but instead a ranking

[7] Since the fact situation contains only assignments of values to dimensions, and the outcome is just that, a possible perspective is that of changing the entire representation of a case by no longer separating fact situation and outcome, see [VGPV23]. For continuity, we will keep the separation.

of applications by how suitable for adoption their cases are. This means, that higher scores in the evaluation mean a higher place on the waiting list, and an earlier chance for adoption. The outcome dimension ω then represents what we will call the *suitability score*, with higher values meaning more suitable and lower values less suitable. This approach is similar to that used for example by universities that grade applicants to determine the best fit. We could also include a threshold which applicants need to reach in order to get placed on the list in the first place. Applications that get a suitability score under this threshold would be considered denied.

Using this new scenario, let us consider the case base containing the cases of Maria and Linus from Example 15. Their cases are now only represented by their fact situations

$$F_m = \{(sqm, 60), (pet, cat), (income, 800), (work, regular)\}$$

for Maria and

$$F_l = \{(sqm, 90), (pet, dog), (income, 1200), (work, home)\}$$

for Linus, as well as their suitability score. Suppose that Maria' case is

$$m = \langle F_m, (\omega, 10) \rangle$$

meaning she has a suitability score of 10, while Linus' case is

$$l = \langle F_l, (\omega, 30) \rangle$$

with a suitability score of 30. Recall that the higher score for Linus means that he is deemed more suitable for adopting a dog than Maria. How the shelter chooses to scale the values and how it reaches the exact value is not relevant for our examples. In practice, every institution that makes the final decision would probably use some specific and individual system of values.
To see how the model constrains new decisions, we first look again at the fact situation of Oscar, from Example 15. To recall, the fact situation is

$$F_o = \{(sqm, 80), (pet, cat), (income, 900), (work, shifts)\}$$

As we already observed before when comparing Oscar's fact situation to that of Maria, we see that while in some dimensions the values in Oscar's fact situation favor the defendant more, this is not the case in all dimensions. In the complete rule model, which is a reason model, we were still able to see that the model forced a decision. The hierarchical model however is a result model, which only supports a fortiori reasoning. That means, that according to the hierarchical model, the suitability score of Maria's case is no upper bound for the suitability score of Oscar's case. Obviously, Oscar's fact situation is also not at least as strong for the plaintiff as Maria's, so her case's suitability score is also no lower bound. As for the constraint from Linus' case, we can see that indeed, for all dimensions, the values in Oscar's fact situation are more favorable for the defendant than in Linus' fact situation. This means, that the court has an upper bound on its decision, it has to assign ω a value less or equal to that of Linus' case.

This concludes all result and reason models from the literature that we present in this work. We will provide a summary in Section 2.9, but before that we take a look at two connections of the reason model to formal logic.

2.8 Logic and the reason model

Having now seen a variety of precedential reasoning models developed for legal reasoning, we move on to connecting them to formal logics. For this, we will focus on the basic, factor-based reason model. We will point out two connections, first the deontic logic that the reason model creates [Canng], and then a logic developed for binary classifiers [LL23] that can be used to formulate the reason model in a modal logic [LLRS22].

We will consider deontic logic mostly as a tool giving a different perspective on the reason model. However, future work may involve looking at the deontic logic of other reason models, in particular those with a many-valued outcome like the hierarchical model [vWGPV23], as the many-valued outcome could be used to obtain graded deontic operators. This would be especially interesting for applications of some generalized reason model in a larger normative reasoning system that involves other frameworks alongside it. These investigations are outside of the scope of this thesis.

The modal logic encoding is very useful to tie the reason model to computational logic and AI. Not only does it open the model up to results or techniques developed for modal logic, it also enables a formal procedure for obtaining explanations from the reason model, as shown in [LLRS22]. Providing explanations is a crucial step in building a trustworthy and transparent system for normative reasoning. If our new model is to be used in general settings, having this option to explain the decisions using a modal logic encoding would be a big benefit.

2.8.1 Deontic logic

The connection of the reason model to deontic logic is not hard to see. Deontic logic deals with obligations and permissions which are implicitly present in the language we have used to describe the constraint the models impose. We will present the definition of the deontic logic created by the reason model from [Canng], as well as the results the author mentions.
We again use the two outcomes π and δ. Our language consists only of the two outcomes and the deontic operators for obligation and permission, which are defined based on precedential constraint. To say that an obligation or permission follows from a case base we will use the symbol $\Vdash$. We can then write constraints using this language.

Definition 34 (Deontic operators)**.** Given a case base Γ and a fact situation F, we define permission:

$$\Gamma \Vdash P_F(s)$$

to hold if and only if given the case base Γ the court is permitted to decide the fact situation F for side $s \in \{\pi, \delta\}$ based on the definition of precedential constraint of the factor-based reason model (Definition 13).
The same way we can define obligation,

$$\Gamma \Vvdash O_F(s)$$

to hold if and only if given the case base Γ the court is forced to decide the fact situation F for side s based on the definition of precedential constraint of the factor-based reason model (Definition 13).
We define

$$\Gamma \not\Vvdash P_F(s)$$

and

$$\Gamma \not\Vvdash O_F(s)$$

to hold when permission or obligation respectively do not hold.

From this definition, we can see that the usual duality of permission and obligation is also present in this deontic logic. We state these observations without proof, for the proofs see [Canng].

Observation 1. *Let Γ be a case base, and F a new fact situation. The the following two statements hold:*

1. $\Gamma \Vvdash P_F(s)$ *if and only if* $\Gamma \not\Vvdash O_x(\overline{s})$
2. $\Gamma \Vvdash O_F(s)$ *if and only if* $\Gamma \not\Vvdash P_x(\overline{s})$

Using this definition, Canavotto then formulates one relevant property of the logic and thus the reason model itself, that is that the deontic logic is conflict free.

Observation 2. *Given a case base Γ and a new fact situation F, it is impossible that both $\Gamma \Vvdash O_x(\pi)$ and $\Gamma \Vvdash O_x(\delta)$ hold at the same time.*
Further, it holds that exactly one of the following holds:

1. $\Gamma \Vvdash O_x(\pi)$
2. $\Gamma \Vvdash O_x(\delta)$
3. $\Gamma \Vvdash P_x(\pi)$ *and* $\Gamma \Vvdash P_x(\delta)$

In her work, Canavotto does not investigate this deontic logic further. Given the fact that it is a very restricted logical language this is not surprising. However it does provide a formal context to reason about the constraint of the reason model, and might prove useful in the future. Especially with a many-valued outcome, we might be able to define a deontic logic with graded deontic operators, which would be useful for establishing priorities among norms, or dealing with conflicting norms.

Next, we present a modal logic for binary classifiers and show how the reason model can be encoded using it.

2.8.2 Binary classifier logic

Establishing a connection between the reason model and formal logic is a very natural and desirable step towards using models of precedential reasoning in AI applications. Such a connection was created by Lorini et al. in [LLRS22]. In their work, they reference a logic for binary classifiers from [LL23], and use it to model Horty's factor-based reason model from [HB12] in a modal logic.
In this section, we will first present the logic they use, called *Binary Classifier Logic*, and then show how they translate the reason model into this logic.

2.8.3 Syntax and Semantics of BCL

We begin with Binary Classifier Logic from [LL23], or *BCL* for short. It is introduced as a general logic used to model the behaviors of binary classifiers. Those are models that take a binary feature vector as input, and output a binary classification. Examples of classifiers with binary output include medical tests (disease present or not) or quality control (part meets specification or not).

The classifier structure that BCL models consists of a set of input features, which are then classified by some undetermined classification function. The way this is achieved is by defining a modal logic with slightly modified semantics. The semantics is still close to the standard Kripke semantics for model logic however, as there are still certain states that represent different inputs.
There are some important aspects of BCL and the semantics that we will mention. The first is that the classification function is indeed a function. This means that when a model contains states where the same features are satisfied, then the classification of these states must also be the same.
Additionally, since BCL is a general logic it does not contain any actual classification procedure. That means that for some given input state, there is nothing in the logic itself that determines the classification of the state. This is achieved by requiring the models satisfy some constraints defined for the specific application of BCL, in this case precedential reasoning. We will first present the BCL framework with its language and the semantics, and then we will show how the factor-based reason model can be encoded using BCL. For the presentation of syntax and semantics, we will follow the original definitions and notations in [LL23].

The language of BCL consists of atomic propositions Atm_0 that represent the features and the decision Dec that represents the classification of a given input. For the decision they provide three possible values $Val = \{0, 1, ?\}$, and then write

$$Dec := \{t(x) : x \in Val\}$$

The three options correspond to a binary classification with one unknown outcome for undetermined inputs.
The formulas of BCL are then defined inductively as either

1. p for $p \in Atm_0$
2. $t(x)$ for $t(x) \in Dec$

for the atomic formulas, and for A, B arbitrary BCL-formulas and X a finite subset of Atm_0, we have

$$\neg A \mid A \wedge B \mid [X]\, A$$

as formulas. The $[X]$ operator is a parameterized version of the standard modal $\Box$ operator. For our purposes we do not need to consider the details of this parameterized form as we only consider formulas that use $[\emptyset]$ which acts like the standard S5 modal $\Box$ operator. For a detailed discussion of why the parameterized version was chosen see [LL23]. We will use $\langle\emptyset\rangle$ as the dual to $[X]$, as is common in modal logic.

The semantics of this logic is given in the form of classifier models. A classifier model C consists of a pair (S, f), with S being a set of states (or input instances) and f a decision or classification function, mapping elements from S to the values $0, 1, ?$. The class of classifier models is called CM.
A formula of BCL is interpreted with respect to a pointed classifier model, which is a pair (C, s), where $C = (S, f)$ a classifier model, and s a state in S. This is done using a satisfaction relation $\models$, which is defined as follows.

Definition 35 (Satisfaction of BCL formulas)**.** Given a pointed classifier model (C, s) we have

$$\begin{aligned}
(C, s) \models p &\iff p \in s \\
(C, s) \models t(x) &\iff f(s) = x \\
(C, s) \models \neg\varphi &\iff (C, s) \not\models \varphi \\
(C, s) \models \varphi \wedge \psi &\iff (C, s) \models \varphi \text{ and } (C, s) \models \psi \\
(C, s) \models [X]\,\varphi &\iff \forall s' \in S: \text{ if } (s \cap X) = (s' \cap X) \text{ then } (C, s') \models \varphi
\end{aligned}$$

From this definition we can also see that in fact, $[\emptyset]$ is an S5 modality, as every state shares its intersection with $\emptyset$ with every other state.
We say that a BCL formula φ is satisfiable relative to CM if there exists a pointed classifier model (C, s) such that $C, s \models \varphi$. A formula φ is valid if $\neg\varphi$ is unsatisfiable. Additionally, a formula φ is valid in a classifier model $C = (S, f)$, noted $C \models \varphi$ if $C, s \models \varphi$

holds for all $s \in S$.

One helpful shorthand the authors define is

$$cn_{X,Y} := \bigwedge_{p \in X} p \wedge \bigwedge_{q \in Y \setminus X} \neg p$$

for two subsets $X \subseteq Y \subseteq Atm_0$, which encodes what they call a valuation on Y. It expresses (part of) an input where all features in X are present while all features in Y but not in X are absent.

The authors go on to discuss some properties of their logic, as well as providing an axiomatization along with soundness and completeness proofs. For our purposes, these results are not very relevant, so we will now move on to the translation of the original reason model using the logic we just presented.

2.8.4 Translating the reason model into BCL

We present the translation of the original reason model from [HB12] into BCL as it was defined in [LLRS22]. The main motivation for us to consider this translation lies in the explanations that Lorini et al. define based on their translation. As we mentioned above, the ability to explain the decisions is a huge step towards a trustworthy and transparent system. In order to understand why the translation of the reason model into BCL makes sense and how it is accomplished we first need to consider two observations.
The first is one we already made in Chapter 1, and it is is that the factor-based reason model does in fact behave like a binary classifier. The factors are the binary input features, and the courts decision is essentially a classification.
The second observation is that the concept of precedential constraint as it is defined in [HB12] is captured by the concept of two-way monotonic boolean functions. To see this, we can consider the a fortiori constraint of the result model. A new case is forced to be decided for the plaintiff if and only if it contains at least the pro-plaintiff factors of a precedent and no additional pro-defendant factors compared to the precedent. Both of these conditions together express a sort of monotonicity of the decisions, where the decision is monotonic with respect to the pro-plaintiff factors and anti-monotonic with respect to the pro-defendant factors. This connection between precedential constraint and two-way monotonicity will be at the center of the translation of the factor-based reason model, as well as later for the new translations we define for other reason models. This two-way monotonicity will be expressed in BCL to translate the precedential constraint into a logical formula. On top of that, we need a formula that guarantees that our model actually provides classifications for all possible fact situations, which is also encoded in a formula. Without such a formula, we could get models that simply contain all the precedent cases, but that do not contain any additional states, making them useless to obtain information on constraint. These two structural formulas for a complete classification as well as the two-way monotonicity define the general class of classifier models that represent precedential reasoning.

Translating individual cases and case bases is then simply specifying that certain inputs must have the correct classification as it is in the case base.

Before we can define the structural formulas we need to add one assumption to our language of BCL, and that is that the atomic propositions that represent the factors can be split into disjunct sets of factors that are pro-plaintiff and pro-defendant. We will call these sets Atm_0^π and Atm_0^δ respectively.
The final change is purely in notation. Since we use s to denote an arbitrary outcome, and in BCL s is used to denote a state in the set S that is part of a classifier model, we need to change one of the two names. Since we have used s to mean an arbitrary outcome for the entire work so far, we will change how we denote an arbitrary state of a classifier model, instead now using a. We will also use our symbols π and δ for the outcome instead of 1 and 0.

The two structural formulas we mentioned are defined as follows. The formula that guarantees a complete classification is called `Compl` and is defined as

$$\texttt{Compl} := \bigwedge_{X \subseteq Atm_0} \langle \emptyset \rangle \, cn_{X,Atm_0}$$

while the formula for two-way monotonicity is called `2Mon` and is defined as

$$\texttt{2Mon} := \bigwedge_{s \in \{\pi,\delta\}, X \subseteq Atm_0^s, Y \subseteq Atm_0^{\overline{s}}} \Big((\langle \emptyset \rangle \, cn_{X \cup Y, Atm_0} \wedge t(s)) \rightarrow \bigwedge_{Atm_0^s \supseteq X' \supseteq X, Y' \subseteq Y} [\emptyset] \, (cn_{X' \cup Y', Atm_0} \rightarrow t(s)) \Big)$$

which expresses exactly the intuition we outlined: For all possible fact situations, if the fact situation was classified to have some outcome s, then we get that for all other fact situations, if they satisfy at least all the pro-s factors, and at most all the pro-$\overline{s}$ factors, then those fact situations must also be classified to have outcome s.

All classifier models that satisfy both `Compl` and `2Mon` make up the class $\mathbf{CM}^{prec}$ for classifier models that satisfy the theory of precedential constraint.

$$\mathbf{CM}^{prec} := \{C = (S, f) \in \mathrm{CM} \mid \forall a \in S : C, a \models \texttt{Compl} \wedge \texttt{2Mon}\}$$

Satisfiability and validity for $\mathbf{CM}^{prec}$ are defined as for CM.

What remains is the translation of an actual case base into a BCL formula, such that any classifier model in $\mathbf{CM}^{prec}$ that satisfies the formula is an accurate representation of the case base. For this we first define a formula for an individual case. Note that in [LLRS22] the authors first define a translation for the result model, which we omit. The translation is defined as follows.

Definition 36 (Translation of a case, factor-based reason model)**.** We define a translation function tr that maps a case $c = \langle F, r, s\rangle$ in the format of the factor-based reason model to a BCL formula. Recall that we can split the fact situation into the pro-plaintiff factors F^{π} and the pro-defendant factors F^{δ}.

$$tr(\langle F, r, s\rangle) := \langle\emptyset\rangle\, (cn_{Premise(r)\cup F^{\overline{s}}, Atm_0}) \wedge t(s)$$

and naturally the extension for an entire case base

Definition 37 (Translation of a case base, factor-based reason model)**.** Let Γ be a case base of the factor-based reason model. We then define

$$tr(\Gamma) := \bigwedge_{\langle F,r,s\rangle\in\Gamma} tr(\langle F, r, s\rangle)$$

With these two definitions in place, the authors then show that they in fact achieve the encoding of the reason model in the logic. Note that because the models all classify every possible fact situation due to `Compl` there is no mention of constraining new cases, as each new case has already been classified, either due to some constraint according to `2Mon` or just by some assignment. What the model therefore captures is that if there is a classifier model in $\mathbf{CM}^{prec}$ that satisfies the formula obtained by translating the case base then the case base must be consistent and vice versa. The notion of consistency is here the one defined for the factor-based reason model (in this work it is Definition 12). The equivalence is expressed in the following theorem.

Theorem 1. *Let Γ be a case base of the factor-based reason model. Then Γ is consistent if and only if the formula $tr(\Gamma)$ is satisfiable in the class $\boldsymbol{CM}^{prec}$.*

An important note is that we can also phrase this in terms of precedential constraint, which is simply a corollary of this theorem.

Corollary 1. *Let Γ be a consistent case base of the factor-based reason model and F be a new fact situation. Then the court is permitted to decide F for outcome s using rule r if and only if the formula*

$$tr(\Gamma) \wedge tr(\langle F, r, s\rangle)$$

is satisfiable in $\boldsymbol{CM}^{prec}$.

We will not repeat the proofs, however we will provide an example of a translated case base that uses the setting of our running example.

Example 17. To see how the translation looks for an actual example, we consider the case base from Example 4 which contains the cases of Ursula and Victor. To recall, those cases are

$$u = \langle \{smallApp, prevPet, suffInc\}, \{suffInc\} \to \pi, \pi\rangle$$
$$v = \langle \{prevPet, suffInc, shifts\}, \{shifts\} \to \delta, \delta\rangle$$

To translate them into BCL, we first need to fix the set of atomic propositions. We will keep the names of the factors for the atoms, to maintain the readability. For the decision, we will stick with $t(\pi)$ for a decision for the plaintiff, and $t(\delta)$ for a decision for the defendant, as we have used it in the definitions.
The formula for the case u is then

$$tr(u) = \langle\emptyset\rangle\,(suffInc \land smallApp \land \neg prevPet \land \neg shifts \land t(\pi))$$

since we need to include as positive atoms the premise of the rule, $suffInc$ and all factors for the other side than the outcome, so $smallApp$, and as negative atoms all remaining atoms. For the case v we get

$$tr(v) = \langle\emptyset\rangle\,(shifts \land prevPet \land suffInc \land \neg smallApp \land t(\delta))$$

with the same procedure.
From these two cases, we can see how we can apply the formula `2Mon` to get information on new cases. In particular, we will look at how it applies precedential constraint to new cases. For this, we consider the case of William from Example 6 with the fact situation

$$F_w = \{shifts, smallApp, prevPet\}$$

To apply the corollary, we need to create a potential decision of this case, and then check whether adding the formula that represents the potential new case to our existing formula of the case base results in a satisfiable formula.
Since we know from Example 6 that precedential constraint forces a decision for the defendant, we will see how this presents itself in the BCL translation. We do so by considering a potential decision for the plaintiff, and seeing why it is unsatisfiable. The only possible decision for the plaintiff would be using the rule $\{prevPet\} \to \pi$, which when translated would result in the formula $\langle\emptyset\rangle\,(prevPet \land shifts \land smallApp \land \neg suffInc \land t(\pi))$. Now, to see how the resulting formula is inconsistent, we just need to consider a classifier model that satisfies all three formulas that represent the three cases of Ursula, Victor and William, and then see that the formula `2Mon` is not satisfied. For this, we look at the relevant clause in the `2Mon` formula, which is

$$\begin{aligned}&\langle\emptyset\rangle\,(shifts \land prevPet \land suffInc \land \neg smallApp \land t(\delta)) \to\\ &\qquad\qquad [\emptyset]\,(shifts \land smallApp \land prevPet \land \neg suffInc \to t(\delta))\end{aligned}$$

To see that this is in fact one of the clauses in `2Mon` we can simply verify that indeed,

$$\{shifts, prevPet, suffInc\}$$

is a subset of all atomic propositions, and that

$$\{shifts, smallApp\}$$

is a superset of the pro-defendant atoms, while $\{prevPet\}$ is equal to the pro-plaintiff atoms. That means that if there is a state where the three atoms $shifts, prevPet$ and

$suffInc$ are satisfied while the atom $smallApp$ is not satisfied, then for any state where the atoms $shifts, smallApp$ and $prevPet$ are satisfied while the atom $suffInc$ is not satisfied, that state must also satisfy $t(\delta)$.
However, as we can see from the three formulas, we do indeed have a state that satisfies the premise, and a state that satisfies exactly those atoms as in the conclusion. However, the outcome is wrong, and since each state satisfies exactly one of the outcomes, the formula `2Mon` is not satisfied. The translated case base along with the potential decision is in fact not satisfiable in $\mathbf{CM}^{prec}$ and therefore the court cannot decide the case for the plaintiff.

Verifying that the court can indeed decide the case for the defendant is an easy exercise that we skip at this point.

Based on this translation, Lorini et al. then show how to obtain three types of explanations for the decisions of the reason model, namely abductive, contrastive and counterfactual explanations. We will not go into the technical details of how these explanations are obtained, we are mostly interested in the fact that there are established ways to use a BCL encoding of a precedential reasoning model to provide such explanations.

2.9 Models for precedential reasoning: a summary

Now that we presented and discussed some models for precedential reasoning, we provide a summary of the models and features we discussed, to identify which aspects we need to carry over into the new model to achieve the four desiderata we presented in Chapter 1. We see in Table 2.1 all models that we have presented in this chapter, along with how they represent the case information, what method of reasoning they use, and what form the outcome has. Beginning with the fact representation, in Chapter 1 we discussed the need for an expressive representation of facts. Using only factors like the original reason model will not give us the desired expressive power. Factors work well for legal reasoning, because courts will often look for the same patterns in the facts, and identify established legally relevant aspects[8]. Using factors to represent the presence or absence of these aspects makes sense in an environment where the interpretation of each factor is done by an expert. In a general scenario however it might not be possible to abstract the case information in such a way. Representing the facts directly, for example by using a value, offers a better solution. For this reason we will use dimensions to represent the case information, like we have seen in the models of Horty [Hor20] and Prakken [Pra21]. The second aspect we presented in Chapter 1 relates to the reasoning principle. While the result model with its a fortiori reasoning certainly provides a sound and intuitive reasoning method, it is a weaker constraint than what we are looking for, as it only considers cases that are stronger in every aspect. For a system in a general context, being

[8] For an example, consider the domain of US Trade Secret law, which was formalized using factors for the HYPO and CATO systems [RA87, AA97]. Aspects like whether the information was reverse engineerable have a legal significance to the court case, and are thus used as a factor.

able to obtain constraint about more nuanced situations is certainly desirable. Using explicit justifications in the form of rules allows us to obtain more normative information from the cases, while keeping the reasoning mechanism intuitive. Thus, we will use the reason model's form of constraint as in the models in [Hor20, Pra21].

Moving on to the outcome it is obvious that a binary outcome is not suitable for general AI applications. While binary classifiers are very common in AI applications, normative reasoning includes many challenges that cannot be addressed by a binary outcome. As discussed in Chapter 1, a binary classifier fails for example to choose the lesser of two evils, or to identify the better of two good solutions. It also forces a strict line between the two outcomes, instead of a potentially gradual boundary. For this reason, we will include the outcome dimension approach we took from [VGPV23].

Finally, as we argued before, generalizing the model to work with inconsistent case bases is a necessary step towards usability of the model. We will therefore follow the ideas of [Canng] to adapt the new model we define to enable reasoning based on inconsistent case bases.

Having identified these aspects of the previous models, we now move on to the definition of our new model.

Table 2.1: Overview of precedential reasoning models

	Fact representation	**Reasoning**	**Outcome**
[HB12]	Factors	Reason	Binary
[Canng]	Factors	Reason	Binary
[Hor20]	Dimensions	Reason	Binary
[Pra21]	Dimensions	Reason	Binary
Adapted from [VGPV23]	Dimensions	Result	Many-valued

CHAPTER 3

Our model

Having seen how previous models for precedential constraint work as well as how they connect to formal logic, we will now present our own model. We will build our model based on the existing models in the way we just described in Section 2.9.To recall, we concluded that we will be using a reason model based on dimensions, combining it with the many-valued outcome of the hierarchical approach. Additionally, we want to allow for reasoning based on inconsistent case bases, as the assumption of a consistent case base is unrealistic in an AI setting. Combining these aspects into a new model is a step towards a precedent based reasoning model applicable to more general settings, as opposed to purely legal applications.
There are three steps we need to take to obtain our new model. The first is to decide which of the two dimension-based reason models we want to follow. In Chapter 2 we have seen the approach of Horty, which uses magnitude factors to build a reason model for dimensions, as well as the approach of Prakken, which uses a complete rule that simply assigns bounds on every dimension to justify the outcome. We will follow the approach of Prakken, and to see why we present a formal comparison of the two models in Section 3.1. The second step is to combine the two concepts of a dimension-based reason model and the dimensional outcome from the hierarchical model. This step requires some modifications to the case representation and the reasoning principle. We motivate these modifications and define our new model in Section 3.2. We proceed as usual, presenting the knowledge representation followed by the constraint, and motivate our decisions with examples. In Section 3.3 we generalize the model we introduce to allow reasoning based on inconsistent case bases.
The result is a dimension-based reason model with a dimensional outcome that is able to constrain decisions based on an inconsistent case base. The model addresses the demands we formulated in Chapter 1.

On top of that we introduce translations of two of the models we showed in Chapter 2 into BCL, following the work in [LLRS22].

3.1 Comparison of two dimension-based reason models

The two dimension-based reason models, described in Chapter 2, present some clear similarities. Both require the fact situation to assign a value to every dimension. Also, both allow the court to specify a reference value for a dimension that provides a justification for the decision.

The main difference is in the way they define constraint. Prakken's complete rule model [Pra21] compares the new fact situation directly with the rules in the case base, and applies constraint whenever a rule applies. Horty's magnitude factor model [Hor20] reuses the idea of a preference relation on reasons from the factor-based case, and adapts it to the dimensional case by introducing a new type of factor based on values of dimensions in the form of magnitude factors.

Both models end up with a form of constraint more expressive than a fortiori constraint, which is why we call both models reason models instead of result models. Since they use a similar approach of setting reference values, both use rules based on these reference values and are both stronger than a fortiori constraint, one could assume that they have essentially the same expressive power. However, it turns out that the models in fact differ when it comes to the constraint they impose on courts. This difference is somewhat surprising, but we find that it is more related to the flaw of Horty's original dimension-based reason model from [Hor19] that we discussed in Section 2.5, than to the definitions of constraint[1].

Since the main difference of the models is arguably just an oversight in the definition of one of the models however, this suggests that both authors have a similar idea to how a justification for a decision should look in the dimensional case.

Still, to the best of our knowledge a formal comparison between the two models has not been done in the literature before. Hence this is the first original contribution of this thesis.

We will show two things:

1. If we convert a case base from the complete rule model into a case base using the magnitude factor model, the two case bases are equivalent.

2. If we convert a case base from the magnitude factor model into a case base using the complete rule model, the case bases are not equivalent.

These two statements require some additional explanations. First of all, what do we mean by converting a case base of the one model into the other. And second, what do

[1] The difference arises from the fact that in the magnitude factor model, the court does not need to provide reference values to all dimensions. This creates a situation, where a later court might use an unused dimension in a way similar to the flaw of the original magnitude factor model we showed in Example 13 to justify a decision that intuitively should not be permitted.

we mean by equivalence. The second point can be dealt with quite simply, by defining the equivalence based on the constraint the case bases impose.

Definition 38 (Case base equivalence)**.** Given two case bases Γ and Σ (with potentially different representations), we say that Γ and Σ are equivalent and denote it by

$$\Gamma \equiv \Sigma$$

if and only if for every new fact situation F, whenever Γ forces a decision for outcome s then so does Σ and vice versa (using the definition of constraint that matches the representation).

This definition of case base equivalence is quite intuitive, and will allow us to compare case bases that use different representations.

The other problem, that of converting a case base of one model to another is less obvious, at least in one direction. Finding a good conversion is of course crucial to the results we claim, so we will provide explanations for why we claim the conversions allow for a true comparison of the two models.
For the direction of converting a complete rule model case base into a magnitude factor model case base we argue that our conversion is the only reasonable approach. This is because we can use one magnitude factor for each of the dimensions with the same reference value as in the rule of the original case, and use all of these magnitude factors in the rule of the converted case.

Definition 39 (Magnitude factor conversion)**.** Let $c = \langle F, r, s \rangle$ be a case using the complete rule model representation, with a dimensional fact situation F, a rule r that contains a value assignment for each dimension and s one of the two outcomes π and δ. We define the magnitude factor conversion of c

$$mfConversion(c) := \langle F', r', s' \rangle$$

to be the case in the magnitude factor model representation where the fact situation F' is identical to F, the outcome s' is identical to s and for every dimension d in the domain D there is a magnitude factor in the premise of r' that favors the outcome s' and that has a reference value that is identical to the value assigned to d by the rule r.

a) $F' = F$

b) $s' = s$

c) $r' = \left\{ M^{s'}_{d,r(d)} \mid d \in D \right\} \to s'$

The magnitude factor conversion of a case base Γ is then the case base

$$\Sigma := \{mfConversion(c) \mid c \in \Gamma\}$$

This definition is not surprising and clearly provides an intuitive translation of a case base in the complete rule model to a case base in the magnitude factor model. We can observe that if c is a case that uses the complete rule model representation, then $mfConversion(c)$ is a valid case that uses the magnitude factor model representation. One important aspect to note, is that the converted cases rule contains a magnitude factor for each dimension, so in a way it has a complete rule.

For the other conversion, we do need to take a less obvious approach. To obtain a case with a complete rule, every dimension needs to have a reference value, however that is not required by the magnitude factor model. As we mentioned above (see Section 2.5), it is this missing requirement that leads to an unintuitive property similar to the flaw in Horty's original model.
To allow us to still compare the two models in this direction however, we need to define some conversion. We present what we argue is a reasonable conversion that sets the reference values of the dimensions without a magnitude factor simply to the value that the fact situation assigns the dimension[2]. For all dimensions that did have a magnitude factor in the original rule, we can simply use the same reference value.

Definition 40 (Complete rule conversion)**.** Let $c = \langle F, r, s\rangle$ be a case using the magnitude factor model representation, with a dimensional fact situation F, a rule r that contains magnitude factors and s one of the two outcomes π and δ. We define the complete rule conversion of c

$$crConversion(c) := \langle F', r', s'\rangle$$

to be the case in the magnitude factor model representation where the fact situation F' is identical to F and the outcome s' is identical to s and the rule r' assigns a dimension d the value p if there is a magnitude factor $M^s_{d,p}$ in the premise of r, or the value $F(d)$ if there is no such magnitude factor.

a) $F' = F$

b) $s' = s$

c) $r' = \Big\{(d,p) \mid \exists M^s_{d,p} \in Premise(r)\Big\} \cup \Big\{(d, F(d)) \mid \nexists M^s_{d,p} \in Premise(r)\Big\}$

The complete rule conversion of a case base Γ is then case base

$$\Sigma := \{crConversion(c) \mid c \in \Gamma\}$$

[2] The idea is that by setting the reference value to the value in the fact situation, we appeal to the idea that for a new case where the original rule applies, if it is stronger in those unused dimensions it would naturally have to be decided for the same outcome already, so adding the additional magnitude factors does not change the intent. For new fact situations with weaker values in the unused dimensions, this also guarantees that the rule can be distinguished, which again seems reasonable.

Having defined equivalence and ways to convert case bases of one model to the other, we can now show our two claims. We begin by showing that the magnitude factor conversion of a complete rule case base is equivalent to the original complete rule case base.

Theorem 2. *Given a case base Γ using the complete rule model representation, and Σ the magnitude factor conversion of Γ, we have*

$$\Gamma \equiv \Sigma$$

For the proof, we rely on one result proven by Horty in his 2019 paper [Hor19]. We refer specifically to Observation 1 of that paper. This observation tells us that if a case base is inconsistent, it must be because the premises of two rules in the case base must be inconsistent. This allows us to focus only on the premises of the rules when arguing about consistency, simplifying the proof massively.

Proof. Let Γ be any case base using the complete rule model representation, and Σ the magnitude factor conversion of Γ.
In order to prove equivalence, we need to prove for any new fact situation F_{new}, that

1. If Γ is not forced to decide F_{new} for any side then so is Σ.
2. If Γ is forced to decide F_{new} for outcome s then so is Σ.

We begin by proving the first statement.
Assume Γ is not forced to decide F_{new} for any side. This is equivalent to Γ being permitted to decide F_{new} for both outcomes. Since Γ is using the complete rule model representation, we apply the definition for constraint for the complete rule model, which gives us that for all cases $c = \langle F, r, s \rangle \in \Gamma$ there exists a dimension $d \in D$ such that

$$F_{new}(d) \prec_d^s r(d) \tag{3.1}$$

We now need to show that for both outcomes there exists a rule r using magnitude factors, such that the resulting case base is consistent. We show that the rule r_{new} with the premise

$$Premise(r_{new}) = \left\{ M^s_{d,F_{new}(d)} \mid d \in D \right\}$$

is a possible rule for outcome s, while with

$$Premise(r_{new}) = \left\{ M^{\overline{s}}_{d,F_{new}(d)} \mid d \in D \right\}$$

we get a possible rule for outcome $\overline{s}$ with the resulting case base being consistent for both rules. We can do an easy sanity check to see why this rule intuitively makes sense by looking at the equivalent rule in the complete rule model. This rule would simply contain all value assignments in the fact situation, and use those exactly as justification for the outcome. And this rule clearly does not contradict any of the existing rules, since

no rule in the case base applies to the fact situation. In other words, the court has not given any information on how a fact situation like F_{new} should be decided because no existing rule applies to it, therefore giving a very explicit rule that uses exactly the values of F_{new} is a safe choice.
To prove this intuition, we appeal to the observation in [Hor19]. In particular, we show that for every rule r_s in Σ with outcome s, the premises of r_{new} and r_s are consistent, as well as for every rule $r_{\overline{s}}$ in Σ. We prove the first case, with the second being analogous. To make the proof a bit easier to follow, we will now fix the outcome of our new rule to one of the two outcomes. Suppose then that r_{new} is a rule with outcome π, and let $c = \langle F, r_\delta, \delta \rangle$ be any case in Σ with outcome δ. Using our statement 3.1 from above we get that for c, there must exist a dimension d' such that

$$F_{new}(d') \prec^{\delta}_{d'} r_\delta(d')$$

We then show that

$$Premise(r_{new}) <_c Premise(r_\delta)$$

does not hold, and thus the two premises must be consistent. For this we need to show that of the conditions for preference, either condition 1 or both 2a and 2b are not met. The conditions in this instance are

1 $Premise(r_\delta) \Vdash Premise(r_\delta)$ and

2a $F \vDash Premise(r_{new})$ or

2b $\overline{Premise(r_\delta)} \Vdash Premise(r_{new})$

Clearly, condition 1 is met. However, condition 2a and 2b are both not met, meaning that we do not have preference.
Towards a contradiction, suppose 2a was met, that means F satisfied every factor in $Premise(r_{new})$, in particular the factor $M^{\pi}_{d',F_{new}(d')}$. That means we have

$$F \vDash M^{\pi}_{d',F_{new}(d')}$$

which by definition gives

$$F_{new}(d') \preceq^{\pi}_{d'} F(d')$$

and by duality of the ordering of the dimensions values

$$F(d') \preceq^{\delta}_{d'} F_{new}(d')$$

However, we know from c being a well-formed case, that F satisfies the premise of the rule r_δ, which must contain a magnitude factor for dimension d'. By definition of the magnitude factor conversion, this magnitude factor must be $M^{\delta}_{d',r_\delta(d')}$, and F satisfying that factor means that

$$r_\delta(d') \preceq^{\delta}_{d'} F(d')$$

This however contradicts the assumption 3.1 on page 57 which told us that

$$F_{new}(d') \prec^{\delta}_{d'} r_\delta(d')$$

which contradicts the ordering we deduced. Thus, F cannot satisfy every factor in $Premise(r_{new})$, and condition 2a is not met.
To show that 2b is not met, we just need to provide a counterexample for

$$\overline{Premise(r_\delta)} \Vdash Premise(r_{new})$$

that is a fact situation Y such that $Y \vDash \overline{Premise(r_\delta)}$ but $Y \nvDash Premise(r_{new})$. Let Y be the fact situation that assigns all dimensions the reference values in r_δ, so

$$Y = Premise(r_\delta)$$

which means that in particular $Y(d') = r_\delta(d')$. Such a fact situation is clearly possible and it trivially satisfies $\overline{Premise(r_\delta)}$.
We then show that Y does not satisfy $Premise(r_{new})$ by showing that Y does not satisfy the factor $M^{\pi}_{d',F_{new}(d')}$. This is an immediate consequence of the observation from above that

$$F_{new}(d') \prec^{\delta}_{d'} r_\delta(d')$$

or by duality

$$Y(d') = r_\delta(d') \prec^{\pi}_{d'} F_{new}(d')$$

In order to satisfy the factor $M^{\pi}_{d',F_{new}(d')}$ we would however need $F_{new}(d') \preceq^{\pi}_{d'} Y(d')$. This shows that condition 2b is not met, thus meaning that the conditions for preference of reasons are not met and we do not have

$$Premise(r^{new}) <_c Premise(r_\delta)$$

which means that the new rule must be permitted.
The proof for the case that r_{new} is a rule with outcome δ is analogous.

We can then move on to proving the second statement, that if Γ forces a decision for s, then so does Σ.
We fix the outcome again, for ease of readability. Suppose that Γ forces a decision for π. By definition, that means that there exists a case $c = \langle F, r, \pi \rangle \in \Gamma$ such that

$$r(d) \preceq^{\pi}_{d} F_{new}(d) \tag{3.2}$$

for all dimensions $d \in D$.
In order to prove that Σ also forces a decision for π, we then need to show that there is no rule r_{new} with outcome δ that leads to a consistent case base. Again appealing to Horty's observation, this means that for every such rule r_{new} there is a rule in the case base such that the two premises are inconsistent. We will show that this rule is the rule r of the case c from the assumption.

Let then r_{new} be a rule with outcome δ such that $Premise(r_{new})$ is satisfied by F_{new}. Let

$$c_{new} = \langle F_{new}, r_{new}, \delta \rangle$$

be the case that results from the court deciding F_{new} for δ using rule r_{new}. We show that the case base $\Sigma \cup \{c_{new}\}$ is inconsistent, by showing

i) $Premise(r) <_{c_{new}} Premise(r_{new})$

ii) $Premise(r_{new}) <_c Premise(r)$

In both cases, we need to show that the conditions for preference of reasons are met. As before, condition 1 is trivial, so we will focus on condition 2a and 2b. For i), we show that condition 2a holds, so that

$$F_{new} \vDash Premise(r)$$

This however is an immediate consequence of our assumption for case c, which is that

$$r(d) \preceq^{\pi}_{d} F_{new}(d)$$

for all dimensions, and therefore F_{new} satisfies every factor in $Premise(r)$ as those are all magnitude factors for side π with reference value $r(d)$.
For ii) we show that condition 2b holds, so that

$$\overline{Premise(r)} \Vdash Premise(r_{new})$$

For this, we need to show that for any fact situation Y that satisfies $\overline{Premise(r)}$, Y also satisfies $Premise(r_{new})$.
Let Y be any fact situation that satisfies $\overline{Premise(r)}$, so Y satisfies every factor is $\overline{Premise(r)}$. Since r has a magnitude factor for every dimension this means that Y must have certain values for each dimension, in particular for those dimensions that have a factor in $Premise(r_{new})$. Let $M^{\delta}_{d,r(d)}$ be any one of those factors from $\overline{Premise(r)}$. Since Y satisfies it, we get $r(d) \preceq^{\delta}_{d} Y(d)$. We need to show, that Y satisfies the factor for dimension d in $Premise(r_{new})$. Let $M^{\delta}_{d,p}$ be that factor, with reference value p. Since $Premise(r_{new})$ is satisfied by F_{new}, every factor is satisfied, including $M^{\delta}_{d,p}$, so we have $p \preceq^{\delta}_{d} F_{new}(d)$. However, from the dual of our assumption 3.2 on page 59 we get

$$F_{new}(d) \preceq^{\delta}_{d} r(d)$$

giving us an ordering like this:

$$p \preceq^{\delta}_{d} F_{new}(d) \preceq^{\delta}_{d} r(d) \preceq^{\delta}_{d} Y(d)$$

Since the ordering is transitive, this gives us

$$p \preceq^{\delta}_{d} Y(d)$$

meaning Y satisfies $M^{\delta}_{d,p}$. Since this holds for all dimensions, Y satisfies all factors in $Premise(r_{new})$ and thus

$$\overline{Premise(r)} \Vdash Premise(r_{new})$$

holds.
Therefore, both

i) $Premise(r) <_{c_{new}} Premise(r_{new})$

ii) $Premise(r_{new}) <_{c} Premise(r)$

hold, and the resulting case base is inconsistent. Therefore, there cannot be any rule r_{new} for δ, and we get that Σ forces a decision for π. This proof can be done analogously for the case that Γ forces a decision for δ. □

This result is not that surprising, as the magnitude factor conversion of a complete rule case base still produces rules that provide reference values for every dimension. However, the formal proof shows, that the notion of preference among reasons based on magnitude factors is not necessary, provided that every rule in the case base is a complete rule. How about the case then, that we allow rules that are not complete. How does a magnitude factor case base compare to its complete rule conversion. As we have already claimed, the case bases are not equivalent. We will provide a counterexample, where the original magnitude factor case base allows both outcomes, while the complete rule conversion forces an outcome. We will then discuss the nature of the counterexample, and what it means for the magnitude factor model.

Theorem 3. *Given a case base Γ using the magnitude factor model representation, and Σ the CR conversion of Γ, we have*

$$\Gamma \not\equiv \Sigma$$

Proof. The claim follows by exhibiting a counterexample. We will use the scenario of the pet shelter as in our other examples, restricting the domain to the dimensions *income* and *pet*. We consider the case base containing only the case of Christian. Christian has an income of 1200, but has not previously owned a pet, so his fact situation is

$$F_c = \{(income, 1200), (pet, none)\}$$

The court decided in favor of the plaintiff, using as a rule

$$r_c = \left\{M^{\pi}_{income,1000}\right\} \rightarrow \pi$$

His case then looks as follows:

$$c = \langle F_c, r_c, \pi\rangle$$

and it is the only case in the case base. To recall, the rule provided by the court argues, that deciding for Christian is justified, because his income is higher than the reference value of 1000. As the court did not include any other magnitude factors in the decision, it arguably says that the fact that Christian has not owned a pet before does not matter for this case, or at least it does not suffice to rule against him.

The complete rule conversion of the case base would then represent Christian's case as follows:

$$c' = F_{c'}, r_{c'}, \pi$$

with $F_{c'} = F_c$, and

$$r_{c'} = \{(income, 1000), (pet, none)\}$$

as we take the magnitude factors reference value when present, and the fact situation value otherwise.

Let us now consider the new fact situation of Bella, who has an income of 1100, and has previously owned a fish. Her fact situation is then

$$F_b = \{(income, 1100), (pet, fish)\}$$

Clearly, her income is less favorable than Christian's, while her experience owning a fish before favors her more than Christian's total lack of experience. Bella's income also is still comfortably higher than the reference value of 1000 provided by the court in the rule of Christian's case. We would then assume, that the court would be forced to decide Bella's case in her favor. And indeed, in the complete rule case base that we get from the conversion, the decision for Bella is forced, as her fact situation presents more favorable values for her in every dimension compared to the values in Christian's rule.

However, we claim that there is a rule against Bella that is consistent with Christian's rule according to the definition of constraint in the magnitude factor model. We will see that it exploits the same flaw as Example 13.

The rule we claim is consistent is

$$r_{new} = \left\{M^{\delta}_{income,1150}, M^{\delta}_{pet,cat}\right\} \to \delta$$

and it says, that deciding against Bella is justified, as her income is less or equal to 1150, and her previous ownership of a fish is less or equally favorable than having owned a cat. This rule is clearly in conflict with the rule in Christian's case, as in his case an income of more than 1000 was sufficient, even with no pet experience at all. We would therefore imagine that it is inconsistent, and cannot be used to justify deciding against Bella.

However, we now argue that the rule is indeed consistent by showing that

$$Premise(r_{new}) <_c Premise(r_c)$$

does not hold, hence there can be no conflicting preferences.

For this, we show that neither condition

2a $F \vDash U$ or

2b $\overline{Premise(r)} \Vdash U$

hold for the two premises, so that neither does Christian's fact situation satisfy the premise of Bella's rule, nor does the inverse of his rule's premise entail that of Bella's rule.
As for 2a, we can see that Christian's fact situation does not satisfy $Premise(r_{new})$, as his income is in fact greater than 1150, meaning that we do not have

$$r_{new}(income) \preceq^{\delta}_{income} F_c(income)$$

This is what gave rise to the flaw in Horty's original model, and what was fixed by introducing condition 2b. However, in this instance condition 2b does not cover this case. To show that 2b also does not hold, we need to show that

$$\overline{Premise(r_c)} \Vdash Premise(r_{new})$$

does not hold. We know that $\overline{Premise(r_c)}$ is a reason for δ containing only one factor $M^{\delta}_{income,1000}$. Let now F be a fact situation, with an income of 900, and previous ownership of a dog. We clearly have

$$F \vDash M^{\delta}_{income,1000}$$

as the income is lower than 1000, however, we do not have

$$F \vDash Premise(r_{new})$$

as we have

$$F \nvDash M^{\delta}_{pet,cat}$$

because in the fact situation X, the experience of having owned a dog is more favorable for the plaintiffs side than that of having owned a cat. This exploit is only possible since the original rule for Christian did not include any magnitude factor for the *pet* dimension, and therefore the reason entailment can be avoided.

So since there is a fact situation that satisfies $\overline{Premise(r_c)}$ but not $Premise(r_{new})$ we can conclude that $\overline{Premise(r_c)}$ does not entail $Premise(r_{new})$ and thus condition 2b is not met.
In summary, this means that the new rule can be used to justify a decision against Bella, without the decision being inconsistent. □

This result clearly mirrors that of Example 13, and should thus be viewed as similarly problematic for the model. It shows, that for a case c decided for side s, as long as its rule does not cover all dimensions, a different court might decide a new case for $\overline{s}$ even though the rule in the case c would tell us intuitively to decide the new case for s. The court however can use the dimensions that are unused by the rule in c as justification, even if the values of the unused dimensions in the new case are stronger for s than in the precedent case.

A conclusion one might draw is that the court should always address every dimension. However, as we have shown, in that case the magnitude factor model and the complete rule model are equivalent, and with the complete rule model using a significantly simpler representation, it seems more practical.

Another possible solution could be to adapt the conditions for preference again, for example by applying the conditions 2a and 2b for each individual factor of a reason instead of the entire reason. This would fix the example we provided, but it is unclear whether it covers all similar issues, and how the resulting constraint compares to other models.

Due to these reasons, we will follow the approach of the complete rule model for the introduction of our own model.

3.2 The new model

Having settled on an approach to dimensions, we move on to present the new model. As mentioned, we combine the complete rule model [Pra21] with the idea of a dimensional outcome as in the hierarchical model [VGPV23]. Before we begin with the definitions of the knowledge representation and constraint, there are a few issues that need to be addressed first. One relates to using a dimensional outcome in a general setting, another to using a rule to enforce a lower or upper bound like in the hierarchical model. One final issue arises when defining constraint based on the representation we will show. We will discuss the first two issues and how we solve them now, and discuss the issue related to constraint after defining our knowledge representation.

1. Issue: Dimensional outcome We mentioned the first issue in Section 2.7 and promised to address it, which we do now. The issue comes from using a dimension as an outcome for scenarios where the outcome is an abstract concept that cannot easily be measured. An example for this could be organ donation candidates, where assigning each patient a score might not be practical. The same could be said for our running example scenario of the pet shelter.

In both these scenarios a concrete value is not necessary, as the only relevant information that subsequent decisions are based on is the ordering of cases. It does not matter whether applicant A got a score of 40 while applicant B only had a score of 35, what matters is that A got a higher score than B.

We therefore introduce a new type of dimension that we will call *case-ordered dimension* which is not a set of values but instead an ordering of all cases in the case base. This case-ordered dimension can then be used as the outcome dimension in scenarios where assigning concrete values is impractical or unnecessary.

We begin by defining case-ordered dimensions and then show how they can be used in an example. It is important to keep in mind that a case-ordered dimension only provides information about a case relative to other cases, and it is therefore only useful in the context of a case base.

Definition 41 (Case-ordered dimension)**.** Let Γ be a case base. A case-ordered dimension d is a linear order $\ll_d$ on all cases $c \in \Gamma$.

Instead of assigning a value for some case-ordered dimension d, a case c in a case base Γ instead has a position in the order $\ll_d$. If we assume that for Γ the case-ordered dimension d is the outcome dimension we write the case $c = \langle F, r, \ll_d \rangle$. To access the position of c in the order $\ll_d$ we give the direct neighbors to either side, which we write as

$$d(c) = (x, c, y)$$

where x is the immediate predecessor to c, so $x \ll c$ and for all $x \ll z \ll c$ we have $x = z$, and y is the immediate successor, so $c \ll y$ and for all $c \ll z \ll y$ we have $y = z$. If a case is the minimum or maximum of the order, we denote this by $d(c) = (c, y)$ or $d(c) = (x, c)$ respectively.

Be reminded that a case-ordered dimension is only useful in the context of a case base, as it only expresses the position of a case relative to other cases. Also, since the ordering of a case-ordered dimension is linear, every case in the case base must be part of the order, meaning that if a new case gets added, the orders of all case-ordered dimensions must be updated.
To see how case-ordered dimensions can be used, we revisit the example of the hierarchical result model, Example 16.

Example 18. We use the same fact situations of Maria and Linus

$$F_m = \{(sqm, 60), (pet, cat), (income, 800), (work, regular)\}$$

and

$$F_l = \{(sqm, 90), (pet, dog), (income, 1200), (work, home)\}$$

but instead of the court deciding a suitability score, it now simply orders the two cases based on who is more suitable. The shelter would then process the adoptions in the order determined by the ranking. We could also add an artificial case into the ranking to symbolize the threshold which needs to be reached by an applicant to be considered accepted.
In this case, the court would provide as an outcome the order

$$m \ll_\omega l$$

This order expresses only that Linus is a more suitable applicant that Maria, without giving any concrete measure of how suitable he is and how big the difference between their suitability is. If there was one available dog, the shelter would allow Linus to adopt it first then.

To see how we would handle the addition of new cases, we follow Example 16 and look at the fact situation of Oscar

$$F_o = \{(sqm, 80), (pet, cat), (income, 900), (work, shifts)\}$$

As we discussed in Example 16, since we are working with a result model for this example, Maria's case does not constrain the decision of Oscar's case, however Linus' does. Since Oscar's case is weaker for the plaintiff than that of Linus, the place of Oscar's case in the outcome ordering is limited by Linus case. This means the court can decide on one of two possible outcome orderings,

$$m \ll_\omega o \ll_\omega l \text{ or } o \ll_\omega m \ll_\omega l$$

to indicate whether Oscar is more or less suitable than Maria.

A generic interpretation of the outcome dimension could be given in terms of our binary outcomes plaintiff and defendant, or π and δ. We could interpret a higher position in the ranking to mean that the case is more in favor of the plaintiff than other cases that are ranked lower. In this way, the case-ordered outcome does not make absolute statements like which case is won by which side, but rather expresses that some cases lean more towards one side than others.

Note that it is possible to implement case-ordered dimensions as we defined them using the standard value-based dimensions. Using an infinite and densely ordered set of values for some dimension d, one could interpret some value assignment not by looking at the actual value but by looking only at how the value compares to other cases. Since the order is dense, it is always possible to assign any new case a value between any pair of cases, thus a new case can be placed in any position in the order. However, using case-ordered dimensions removes the option for misunderstanding the values as relevant by simply removing the values altogether. We therefore argue that they should be used whenever we are interested only in the relative position of a case and not any concrete value.

2. Issue: Rules and dimensional outcome The second issue we need to address relates to adding rules to a dimensional outcome. To better understand this issue we first need to examine the nature of the rules we have seen thus far.
In all previous reason models that had a binary outcome, the rule was used to justify the decision of the court for one of the two sides. This was expressed explicitly in the conclusion of the rule. Moving to a dimensional outcome introduces an issue that we can observe best by considering the constraint of the hierarchical model we presented in Section 2.7.
That model is a result model, therefore it does not have a rule. The way we obtain constraint, by directly comparing two sets of values, is however the same as for the rules

of the complete rule model.
Let us then examine the constraint of the hierarchical result model to observe that a single case can be used to constrain future cases in *both* directions, something that rules cannot.

Example 19. We consider the case base from Example 18, with the cases of Maria and Linus, and with a case-ordered dimension as the outcome. Their fact situations are

$$F_m = \{(sqm, 60), (pet, cat), (income, 800), (work, regular)\}$$

and

$$F_l = \{(sqm, 90), (pet, dog), (income, 1200), (work, home)\}$$

and we assume the outcome to be the ranking

$$m \ll_\omega l$$

Specifically, we will now use the case of Linus to show that the constraint of the hierarchical result model applies in both directions from the same case.

In Example 16 and Example 18 we have already seen how the decision of Linus' case constrains the court in deciding the fact situation of Oscar

$$F_o = \{(sqm, 80), (pet, cat), (income, 900), (work, shifts)\}$$

forcing a decision that must rank Oscar's case lower than that of Linus, due to Oscar's fact situation being weaker in every dimension. Suppose that the court decided on the new ranking

$$m \ll_\omega o \ll_\omega l$$

and now considers the new fact situation of Oscar's friend Olivia with is

$$F_{o'} = \{(sqm, 100), (pet, dog), (income, 1400), (work, home)\}$$

Comparing this fact situation to that of Linus, we can see that now every dimension is stronger for Olivia than for Linus, which means that precedential constraint applies and forces the court to rank Olivia's case higher than that of Linus, leading to the new ranking

$$m \ll_\omega o \ll_\omega l \ll_\omega o'$$

As we can see, the result model allows the same case to be used to force a courts decision towards either side.

For the result model then, this effect does not seem problematic. If a new case is strictly stronger for one side, then its outcome should be constrained towards that side. The fact that this now also applies to the other side does not make the argument unsound.

However, if we now introduce a rule into the scenario, this becomes problematic. Our rules will naturally be used to impose upper or lower bounds on the outcome. A rule in the way we have seen it so far is exclusively used to justify a decision in one specific direction, meaning it would either impose an upper, or a lower bound. This interpretation of a rule in the binary outcome models is a direct consequence of the structure of the model, however we argue that this interpretation of a rule should be upheld in the many-valued outcome scenario. For this we consider an example.

Example 20. Let us use the case of Linus, but from Example 15 again, where we have a case base in the complete rule model. In that setting, we have a binary outcome, and the court decided for the plaintiff. To recall, the fact situation of Linus was

$$F_l = \{(sqm, 90), (pet, dog), (income, 1200), (work, home)\}$$

Since the court decided the case for the plaintiff, it had to assign each dimension in the premise of the rule a value less favorable to the plaintiff than the facts. The rule used in Linus' case was

$$r_l = \{(sqm, 90), (pet, cat), (income, 1000), (work, home)\} \to \pi$$

and does in fact provide a sensible justification of the decision for the plaintiff. It expresses the courts judgment, that any application with at least $90m^2$ apartment size, previous pet ownership experience of at least a cat, and income of at least 1000 and working conditions at least as good as working from home should be decided in favor of the applicant.

Note that the court **does not** express any opinion on what should be done for cases that do not meet that threshold.

Now suppose that in the setting of our new model with many-valued outcome, the court instead used that rule to express the opinion that any applicant that meets this threshold should be ranked at least as high as Linus in a case-ordered outcome dimension. Following the same logic, the court does not want to express any opinion on what should happen with cases that do not meet the threshold.

Suppose now that the court is presented with the new fact situation of Yvonne

$$F_y = \{(sqm, 100), (pet, cat), (income, 1100), (work, home)\}$$

We can clearly see that all values in her fact situation are more favorable for the plaintiff than in r_l. Thus, we would appeal to the rule to justify Linus' case being a lower bound in the outcome ranking for Yvonne's case. This is exactly the intention of the rule.

What happens however, when we consider the new fact situation of Xavier

$$F_x = (sqm, 80), (pet, none), (income, 800), (work, regular)$$

From our intuition, it should be clear that he should be ranked lower than Linus, as he presents a less favorable case.

Keep in mind that the rule the court formulated expressed the opinion, that any case that meets the threshold of the premise should be ranked at least as high as the case specified as the lower bound. It did not intend the rule to provide any information or guidance for fact situations that do not meet the threshold, like the one of Xavier.
It would be natural to assume then, that the rule is irrelevant, and Xavier's fact situation is unconstrained. However, if we compare the values in the rules premise to those in the new fact situation regardless, we could argue that it does apply, and that it now forces an upper bound on the decision. In that way, a rule would act in a similar way to a in the hierarchical result model in that it can be used to justify decisions in both directions.

As we mentioned above, this is not part of what the court wanted to express with the formulation of the rule, and simply a consequence of reasoning with a many-valued outcome.

As we can see, using rules like the ones in the complete rule model breaks with the meaning of a rule we mentioned above.
Our solution for this issue is simple: The court just needs to specify for each rule whether it justifies an upper bound on the outcome or a lower bound on the outcome.
We therefore need two types of rules, which we will call upper- and lower-bound rules respectively. A case therefore can be decided based on a rule that explicitly justifies why the outcome is higher (or lower) than some reference value expressed in the rule. It is even imaginable that the court simultaneously creates an upper and a lower bound, which means our cases may now have two different rules.

Having solved these two issues related to the dimensional outcome and how to combine it with rules we now move on to formally define our knowledge representation. After this we will discuss the third issue that relates to constraint.

Knowledge representation As we discussed in Section 3.1 we will use the complete rule models representation for dimensional fact situations and rules. What changes is that the outcome of a rule is no longer plaintiff or defendant, but rather an upper or lower bound of the outcome dimension. For these definitions, we will assume that the outcome dimension is a standard, value-based dimension, however all of these concepts can be applied to case-ordered dimensions as well. Recall that we use ω to refer to the outcome dimension. The ordering of the dimensions in our domain is understood like in Section 2.7, so if for two values p, q and some dimension d we have

$$p \preceq_d q$$

then we interpret this as "q favors higher values in ω compared to p"[3].
We begin by repeating the definition of a fact situation.

[3] For case-ordered dimensions we would read it as "q favors a higher position in the ordering compared to p".

Definition 42 (Fact situation)**.** Given a domain D, a fact situation F is a set of value assignments of all dimensions of a domain

$$F = \{(d, p) \mid d \in D\}$$

with the requirement that for every pair (p, d), p is one of the values of dimension d.

Now we move on to defining a rule. As we mentioned, for our cases we will end up with two types of rules, but as they are structurally identical, we just provide one definition for a rule, and make the distinction what type a rule is when it is part of a case. The definition is essentially the same as in the complete rule model, only with an outcome value in ω instead of being π or δ. How this outcome value is chosen is a matter of the court that creates the rule, similar to how the court needs to determine the reference values of the premise.

Definition 43 (Rule)**.** A rule r consists of a value assignment for each dimension of a domain, and an outcome value

$$r = \{(d, p) \mid d \in D\} \rightarrow v$$

with $v \in \omega$.

The definition of a case brings the first major innovation that we discussed above. In the case representation, the rules provided or referenced by the court are now explicitly tagged to be either upper- or lower-bound rules. While the definition always contains two rules, it is important to mention again that the court is not required to provide both rules. We will offer a notation to indicate that the court only referenced one rule to justify its decision after the definition. As for the definition itself, specifying how the fact situation, rule and case outcome must match ends up quite cumbersome, however the intuitions are very clear. For the fact situation to satisfy the premise of an upper bound rule, the reference values of the rules premise must naturally be more favorable to higher values than those of the fact situation. The rule is then read as "Since the actual values in this case are lower (favor lower values for ω) than the reference values of the rule, the outcome must be lower than the outcome specified by the rule." The same is true for lower bound rules, just in the other direction. Formally, we therefore define a case as follows.

Definition 44 (Case)**.** A case $c = \langle F, r_<, r_>, (\omega, v)\rangle$ consists of a fact situation F, a lower bound rule $r_<$, an upper bound rule $r_>$ and an assignment of a value v to ω.
The value v must be between the outcomes of the rules $r_<$ and $r_>$, and both rules must apply to the fact situation. This means that each dimension in $\mathit{Premise}(r_<)$ must have a reference value less favorable to a higher outcome than F, and each dimension in $\mathit{Premise}(r_>)$ must have a reference value more favorable to a higher outcome than F. Formally, that means

$$\mathit{Outcome}(r_<) < v < \mathit{Outcome}(r_>)$$

and

$$r_<(d) \preceq_d F(d) \preceq_d r_>(d)$$

for all dimensions $d \in D$.

As mentioned, it is possible for the court to leave out one of the two rules, providing only a lower or upper bound. We will write such a case simply as $c = \langle F, r_<, -, (\omega, v)\rangle$ or $c = \langle F, -, r_>, (\omega, v)\rangle$ respectively.

The definition of a case base is of course again unchanged.
We can now repeat our motivating example for tagging rules and use our definition to make the courts intention clear.

Example 21. We revisit Example 20 and see that introducing tags for lower and upper bound rules makes sure that rules can only be used in the way that represents the courts opinion.
Recall the fact situation of Linus, which was

$$F_l = \{(sqm, 90), (pet, dog), (income, 1200), (work, home)\}$$

We considered that the court wanted to give a lower bound v on the suitability score, and that it justified this with the following rule

$$r_l = \{(sqm, 90), (pet, cat), (income, 1000), (work, home)\} \rightarrow v$$

Now however, the court can explicitly tag this rule as a lower bound rule. We will call this tagged rule $rl_<$. It now carries the explicit judgment of the court, that any application with at least $90m^2$ apartment size, previous pet ownership experience of at least a cat, and income of at least 1000 and working conditions at least as good as working from home should receive a suitability score of at least v.
An since it is a lower bound rule, it gives no information on what should be done for cases that do not meet the threshold of the premise.

In particular, the new fact situation of Xavier that we showed in Example 20

$$F_x = (sqm, 80), (pet, none), (income, 800), (work, regular)$$

is now unconstrained by the rule $rl_<$, and therefore the rule does not give any bound on the decision.

Now we can move on to the third issue we mentioned at the beginning of this section, which is defining constraint based on the case definition we just presented.

3. Issue: Constraint based on this case representation The third issue we need to solve is about the possible gap between the upper and lower bounds imposed by the rules and the actual outcome value the court chooses. In fact, there are two parts of this issue. Both are again unique to the setting with dimensional outcome, as in the binary setting a rule with conclusion π meant the court had to decide the case for the plaintiff. There was no more choice for the court left to do. With a dimensional outcome however, the court has more than two options for the outcome, and is constrained only by upper or lower bounds. Any value that is not ruled out by an upper or lower bound is a permitted outcome.
In the hierarchical model we showed in Section 2.7 this is not much of an issue. The court has to make a consideration based on the fact situation of the case, and is only bound in its decision of an outcome by other cases with strictly stronger or weaker fact situations. Once we introduce rules to impose these bounds however, the situation changes.

The first problem arises when the court decides on the outcome of a new case. If the court is constrained in its decision by some rule r that was created for some precedent case c then the actual outcome value of case c is not considered in the constraint. In fact, all that we know is that if r is an upper bound rule, then the outcome of c is at most the outcome of r, and the outcome for the new case must also be at most the outcome of r. How the two case outcomes relate however is not part of the constraint.

This opens up a big issue, which is that if we only apply constraint based on rules, then we might end up with two cases that go against a fortiori reasoning, so where one case is strictly stronger that the other, but has a lower outcome. To see how such a situation can occur, we consider an example.

Example 22. Suppose we have a case base containing the case of Maria from Example 15. To recall, we use the restricted domain of only three dimensions, $sqm, pet, income$ and $work$. For the sake of an easier representation, we will switch back to using an outcome value. We will again let ω represent a suitability score for our scenario, with higher values favoring the plaintiff[4]. Maria's fact situation is

$$F_m = \{(sqm, 60), (pet, cat), (income, 800), (work, regular)\}$$

We need to provide two rules, and a suitability score to construct a case. Suppose the lower bound rule is

$$rm_{<} = \{(sqm, 50), (pet, fish), (income, 500), (work, regular)\} \rightarrow 5$$

and the upper bound rule

$$rm_{>} = \{(sqm, 100), (pet, dog), (income, 1000), (work, home)\} \rightarrow 40$$

[4] Recall that the suitability score is used to sort the applications in a waiting list, with higher scores meaning priority in adopting a dog. We also mentioned the option of a threshold under which applications are considered denied. For more details, see Example 16

with the suitability score of her case being 10. This gives us the full case

$$m = \langle F_m, rm_<, rm_>, (\omega, 10)\rangle$$

which is the only case in our case base Γ.
Now consider the new fact situation of Naomi. Her fact situation is

$$F_n = \{(sqm, 75), (pet, dog), (income, 900), (work, home)\}$$

and we can observe, that for every dimension Naomi's fact situation is more favorable for her side. Our intuition would then tell us, that a fortiori constraint should force the court to give her a higher score than Maria. In all of the models we have seen so far, a fortiori constraint was either the only principle defined, like in the result models, or it was a consequence of the definition of constraint in the reason models.
However, if we base the constraint in our new model entirely on the bounds of the rules that apply, we can consistently assign Naomi's case a lower score than Maria's case.

We see right away, that both rules $rm_<$ and $rm_>$ apply to Naomi's fact situation. The court is therefore bound in its decision to assign Naomi's case a suitability score between 5 and 40. If this is the only constraint, the court is free to assign her case the score 9, which would result in the following case:

$$n = \langle F_n, rm_<, rm_>, (\omega, 9)\rangle$$

Having both m and n in our case base however is clearly undesired, as Naomi's case is strictly more favorable to her than Maria's case, yet she received a lower suitability score.

This is of course unwanted behavior, so we need to make sure to explicitly include a fortiori reasoning in our definition of constraint.
One practical consequence of this is that rules should ideally be tight bounds, with the actual outcome of the case being very close to the value given by the rule. This reduces the range of values where this problem occurs.

The second problem with the gap between the bounds imposed by the rules and the actual outcome arises when the court creates a new rule for some new case. If we do not restrict the rule selection, and simply require that the new rule applies to the new case and that the outcome of the rule matches with the outcome of the case, then this rule could still contradict existing cases. The new rule could apply to a precedent case, however the outcome value of the precedent might not be within the bound of our new rule. Again we show this situation in practice with an example.

Example 23. Take again the case base containing only Maria's case. The new fact situation we consider is that of Adam.

$$F_a = \{(sqm, 50), (pet, none), (income, 2000), (work, shifts)\}$$

A few things are important to note. First of all, a fortiori constraint does not apply, as we have at least one dimension for each Maria and Adam that favors them more than the other. Also, neither of the rules in Maria's case apply, as the lower bound rule uses the value of $fish$ for the dimension pet, and the upper bound rule uses the value of 1000 for the $income$ dimension. The court is therefore unconstrained in its decision for the suitability score. The court must then also provide new rules to justify the decision. Suppose the court chooses the lower bound rule

$$ra_< = \{(sqm, 50), (pet, none), (income, 1000), (work, shifts)\} \rightarrow 5$$

and the upper bound rule

$$ra_> = \{(sqm, 60), (pet, cat), (income, 2000), (work, regular)\} \rightarrow 8$$

which are both possible rules for Adam's case. The court then decides for a suitability score of 7, thus builds the case

$$a = \langle F_a, ra_<, ra_>, (\omega, 7)\rangle$$

and includes it in the case base.
If we now look at how the new rules interact with Maria's case, we will see a problem. Namely, the upper bound rule $ra_>$ is applicable in Maria's case, but the score in her case is higher than the score of the rule.

The solution for this problem is again straightforward. We simply require that any new rule is consistent with the existing case base, which directly rules out the scenario described in the example we just saw.

With these final issues addressed, we move on to the formal definition of constraint.

Constraint As we explained above, we need to include a fortiori constraint explicitly in our definition. Aside from that, the definition follows the intuition we laid out of imposing an upper or lower bound on the courts decision if there is an upper- or lower-bound rule that applies to the new fact situation.

Definition 45 (Constraint)**.** Given a case base Γ and a new fact situation F_{new}, the value v is a lower bound for the courts decision of F_{new}, if and only if

a) There exists a case $c = \langle F, r_<, r_>, (\omega, v)\rangle \in \Gamma$ such that for all dimensions, the value in F_{new} favors higher values for ω than in F

or

b) There exists a lower bound rule $r_<$ for a case $c \in \Gamma$ such that for all dimensions, the value in F_{new} favors higher values for ω than in $r_<$.

Formally, that means that v is a lower bound for ω if and only if

a) There exists a case $c = \langle F, r_<, r_>, (\omega, v)\rangle \in \Gamma$ such that

$$F(d) \preceq_d X(d)$$

for all dimensions

or

b) There exists a lower bound rule $r_<$ for a case $c \in \Gamma$ such that

$$F(d) \preceq_d r_<(d)$$

for all dimensions.

The definition for v being an upper bound is analogous. The court must choose a value for ω that is greater than the greatest lower bound, and less than the least upper bound. The court is not bound in its decision of F_{new} if and only if there is neither a lower nor an upper bound.

In order to show how this definition is applied we reconsider Example 22.

Example 24. If we again consider the fact situation of Naomi

$$F_n = \{(sqm, 75), (pet, dog), (income, 900), (work, home)\}$$

against the background of the case base containing only Maria's case m, we can see that for Naomi's decision the court now has two lower bounds. First, the lower bound rule

$$rm_< = \{(sqm, 50), (pet, fish), (income, 500), (work, regular)\} \to 5$$

is applicable to F_n, giving the lower bound of 5. However, condition a) now tells us that if there is a case in the case base, that is strictly less favorable than the new fact situation, then suitability score assigned in that case is also a lower bound. This is indeed the case here, since Maria's fact situation is

$$F_m = \{(sqm, 60), (pet, cat), (income, 800), (work, regular)\}$$

so the suitability score of m which is 10 is also a lower bound. Since the court is constrained by the highest lower bound, it now has to set Naomi's suitability score to some value greater than 10.

Finally, we again state the restriction on rule selection, that any new rule must be consistent with the precedent cases in the case base.

Definition 46. Let Γ be a case base of our new model, and let

$$\langle F, r_<, r_>, (\omega, v)\rangle$$

be some new potential decision.
The rules $r_<$ and $r_>$ are permitted if and only if for all cases

$$c = \langle F_c, rc_<, rc_>, (\omega, v_c)\rangle \in \Gamma$$

when $r_<$ applies to F_c then the outcome value v_c is greater or equal to $\mathit{Outcome}(r_<)$ and when $r_>$ applies to F_c then the outcome value v_c is less or equal to $\mathit{Outcome}(r_>)$.

This concludes the definition of our new model for precedential reasoning. To summarize, to obtain our new model we have used the approach to dimensions of [Pra21], using complete rules that assign every dimension a reference value. We have then used the idea of constraining a dimensional outcome with upper and lower bounds from [VGPV23]. In order to combine these two concepts, we needed to introduce new concepts, like explicit a fortiori constraint and two types of rules that can only be used to justify either upper or lower bounds.
Next, we will generalize this model to enable reasoning based on inconsistent precedent cases.

3.3 Reasoning based on inconsistent case bases

Allowing the model to provide meaningful answers even if the underlying case base is inconsistent is a crucial aspect of making the model applicable outside legal reasoning contexts. As discussed in Section 2.4, a real world case base being completely consistent is highly unlikely. The intuitive solution of [Canng] that we presented was to ensure that an already inconsistent case base does not get more inconsistent, meaning that new decisions do not introduce inconsistencies that were not already part of the original case base. To recall, in her work she generalized the original factor based reason model [HB12], where consistency was defined in terms of the preference relation of reasons. The case base not getting more inconsistent meant to gather all pairs of reasons with contradicting preference, and to allow decisions as long as they did not create a new contradicting pair of reasons.
As our model uses the complete rule approach to dimensions we do not have a preference relation on reasons, in fact there is no definition of a consistent case base. In Section 3.3.1 we discuss how inconsistency presents itself in the complete rule model (for the definition of that model see Section 2.6). We will then apply the intuition outlined above to generalize the complete rule model to allow reasoning based on inconsistent case bases. Following that, we adapt the definition to our new model with a dimensional outcome in Section 3.3.2.

3.3.1 The complete rule model with inconsistent case bases

In order to get a better idea of how the intuition of Canavotto's approach to reasoning with inconsistent case bases can be transferred to a dimensional setting that does not use a preference relation on reasons, we will begin by generalizing the complete rule model we presented in Section 2.6. This addresses two steps towards our final generalization, moving to the dimensional setting, and using the rules directly to obtain constraint.

Before defining our generalization however, we first need to clarify what we mean by an inconsistent case base in the complete rule model. The reason models of Horty all use a formal notion of consistency in their definition of constraint. The complete rule model does not define consistency of a case base, because its definition of constraint relies solely on the rules in the case base. However, there are still scenarios that we would describe as inconsistent.

In fact, for a case base of the complete rule model, there are two situations that we will call inconsistent.

1. There could be two rules that can apply to the same fact situation, but that have opposing outcomes. If there is such a pair of rules, a court might be forced to decide a new fact situation for both outcomes, which is impossible.
2. The case base might contain a case that was decided for one side, but there is a rule that applies to the fact situation of that case, which would have forced the other outcome. This means that the decision of the case violates a rule.

Both these situations are clearly inconsistent, and while it seems like they describe two separate scenarios, a closer look reveals that they are equivalent. If there is a case that violates a rule, then that case must itself use a rule that justifies its own outcome. This rule necessarily conflicts with the violated rule. And if two rules are conflicting, then one of the cases the rules belong to must violate one of the two rules. We therefore only need to define inconsistency as the existence of two rules that can apply to the same fact situation but that have different outcomes.

Definition 47 (Inconsistent and consistent case bases, complete rule model)**.** Let Γ be a case base of the complete rule model. Then Γ is inconsistent if and only if there are two cases $\langle F, r, \pi\rangle$ and $\langle F', r', \delta\rangle$ such that the values in rule r are less favorable to the plaintiff than the values in rule r'

$$r(d) \preceq_d^{\pi} r'(d)$$

for all $d \in D$ and it is consistent if and only if it is not inconsistent.

It remains to determine how to define the constraint of a case base that is inconsistent in this way.

To achieve this, we follow the intuition laid out by Canavotto [Canng]. As we discussed before, the idea is to ensure that an already inconsistent case base does not get more inconsistent. As we touched on at the end of Section 2.4, the way this is accomplished is by permitting a decision even if it contradicts a precedent, so long as it is also supported by some other precedent case. In the complete rule model, it is obvious how to capture this idea. A new fact situation that is forced for one side according to some rule may also be decided for the opposite side if there is another rule for that opposite side that also applies.
At the same time, this means that a decision for one side is only forced by some rule if the decision does not contradict a different rule at the same time. This leads to the following definition.

Definition 48 (Constraint of inconsistent case bases, complete rule model)**.** Given a case base Γ and a new fact situation F_{new}, the court is forced to decide F_{new} for side s, if and only if

1. There exists a case $c = \langle F, r, s\rangle \in \Gamma$ such that the rule r applies to F_{new}, so that
$$r(d) \preceq_d^s X(d)$$
for all dimensions $d \in D$.
2. There exists no case $c' = \langle F', r', \overline{s}\rangle \in \Gamma$ such that the rule r' applies to F_{new}, so that
$$r'(d) \preceq_d^{\overline{s}} X(d)$$
for all dimensions $d \in D$.

The court is permitted to decide F_{new} for side s if and only if it is not forced to decide F_{new} for $\overline{s}$.

To see how this definition works in practice, and how it matches the intuition of Canavotto's generalization, we discuss an example.

Example 25. We restrict our domain to a few dimensions for simplicity. In this case, we focus only on the two dimensions *income* and *sqm*. The case base Γ that we consider contains two cases, that of Francesca and Gabriel. Their fact situations are

$$F_f = \{(income, 650), (sqm, 60)\}$$
$$F_g = \{(income, 725), (sqm, 50)\}$$

and the court ruled in favor of Francesca, but ruled against Gabriel. The rules used to justify these decisions are

$$r_f = \{(income, 600), (sqm, 55)\} \to \pi$$
$$r_g = \{(income, 750), (sqm, 60)\} \to \delta$$

We can see right away, that this case base is inconsistent, because the two rules of Francesca's and Gabriel's case contradict each other. In particular, the rule used in Gabriel's case applies to Francesca's fact situation, but the case was decided for the plaintiff despite that.
To see how the definition of constraint works, we now consider the new case of Elena. She presents the fact situation

$$F_e = \{(income, 675), (sqm, 55)\}$$

If we check which rules apply, we see that indeed both existing rules apply to the new fact situation. Usually, this would mean that both outcomes are forced, however the generalized models tells us that in this situation, both outcomes are permitted, because the decision cannot introduce a new inconsistency.
To see how the model still forces decisions, we consider the new case of Ian, with the fact situation

$$F_i = \{(income, 650), (sqm, 65)\}$$

For this fact situation, there is a rule with the outcome π that applies, and since there is no rule with outcome δ that applies (as the value 65 for *sqm* is higher than the value 60 in the rule r_g), the court is forced to decide for the plaintiff.

This concludes the generalization of the complete rule model, which lifts the generalization of [Canng] to the dimensional setting. What remains is the final step, generalizing the new model we defined in Section 3.2 following the same approach we just saw.

3.3.2 Our new model with inconsistent case bases

We now generalize the model we have defined in Section 3.2 to enable it to work with inconsistent case bases. While some adjustments are required to fit the new model, we can draw inspiration from the intuitions and the approach that we presented in the previous section. As we have seen when defining the new model, moving to a dimensional outcome introduces some challenges. The same is true for reasoning based on inconsistent case bases. We will first point out exactly what is different for the new model, and then discuss the intuitions of how we solve them.

As with the complete rule model, we begin by considering which situations we call inconsistent. Similar to the complete rule model, there are two such situations.

1. Similar to the complete rule model, two rules might give conflicting outcomes for the same fact situation. For the new model this occurs, when an upper bound rule forces an upper bound on the decision, but a lower bound rule simultaneously forces a lower bound on the decision that is higher than the upper bound.

2. The other inconsistent situation occurs when a cases decision, so the actual outcome value, violates a lower or upper bound that applies to the fact situation of the case.

> The bound that is violated could either be that of a rule or derived from a fortiori constraint based on some precedent decision.

To have a more concrete idea of these two situations, we will show them with an example.

Example 26. We will again consider the case base Γ of Example 25 that contains the cases of Francesca and Gabriel, as well as the additional case of David. The fact situations of Francesca and Gabriel are as before, so we have

$$F_f = \{(income, 650), (sqm, 60)\}$$

and

$$F_g = \{(income, 725), (sqm, 50)\}$$

with David's fact situation being

$$F_d = \{(income, 800), (sqm, 70)\}$$

The rules need to be adjusted to the new model, as well as the outcomes of the cases. We again use the numerical suitability score as an outcome to allow for a more clear example of the inconsistencies. Suppose the rules used for Francesca's case are now

$$\begin{aligned} rf_< &= \{(income, 600), (sqm, 50)\} \rightarrow 15 \\ rf_> &= \{(income, 700), (sqm, 70)\} \rightarrow 25 \end{aligned}$$

for Gabriel we use the rules

$$\begin{aligned} rg_< &= \{(income, 675), (sqm, 50)\} \rightarrow 30 \\ rg_> &= \{(income, 850), (sqm, 70)\} \rightarrow 35 \end{aligned}$$

and finally the rules for David

$$\begin{aligned} rd_< &= rf_< \\ rd_> &= \{(income, 850), (sqm, 80)\} \rightarrow 40 \end{aligned}$$

Adding a suitability score to each case we get the following three cases

$$\begin{aligned} f &= \langle F_f, rf_<, rf_>, (\omega, 20)\rangle \\ g &= \langle F_f, rf_<, rf_>, (\omega, 35)\rangle \\ d &= \langle F_f, rf_<, rf_>, (\omega, 38)\rangle \end{aligned}$$

From these cases, we can now see the two inconsistent situations. The first is that the upper bound rule of Francesca's case and the lower bound rule of Gabriel's case contradict each other. The rule $rf_>$ argues, that an income of less than 700 with an apartment not larger than 70 square meters means the suitability score must be lower than 25. At the same time, the rule $rg_<$ says that an income over 675 with an apartment greater than 50 square meters needs to receive a score of at least 30. This is a contradiction, which we can see when we consider a new fact situation

$$\{(income, 680), (sqm, 55)\}$$

to which both rules apply. According to $rf_>$ the score cannot be higher than 25, but according to $rg_<$ the score must be at least 30. Any decision the court makes will violate some precedent.

The second situation is that of a bound actually being violated by some case. For this, we take a closer look at the suitability score that was given for the case of David. If we look at the fact situation of David and the upper bound rule of Gabriel's case, we can see that the rule $rg_>$ applies to the fact situation F_d. However, the score in David's case is 38, which is higher than the upper bound provided by the rule in Gabriel's case. This is also clearly an inconsistency.

In the example we can already see the central difference of inconsistency in the new model compared to the complete rule model. While in the complete rule model the two situations were equivalent, in the new model they are not. There can be conflicting rules, but no case that violates any rule, and there can be cases that violate a rule, but there is no rule conflict[5]. This forces us to define inconsistency based on both conditions and to then provide an intuition for how to handle each of the two cases by themselves.

Definition 49 (Inconsistent and consistent case bases, new model)**.** Let Γ be a case base of the new model we defined. Then Γ is inconsistent if and only if at least one of the following holds:

1. There are two cases $c, d \in \Gamma$ with a lower bound rule $r_<$ and an upper bound rule $r_>$ such that there exists some fact situation that both rules apply to and

$$Outcome(r_>) < Outcome(r_<)$$

2. There exists a case $c = \langle F, r, (\omega, v)\rangle \in \Gamma$ and either

   a) there is a rule r' as part of a case in Γ that applies to the F but the value v does not respect the bound of r' or

   b) there is a case $c' \in \Gamma$ such that c is strictly stronger for one side than c', but the value v is weaker for that side than the outcome of c'.

A case base is consistent if and only if it is not inconsistent.

[5] The reader can verify that indeed, while the rules of Francesca's and Gabriel's cases contradict, the outcome values each case was assigned does not violate any of the rules. At the same time, while David's outcome violates a rule, the rules used in his case do not contradict any other rule.

The next step is to find a solution for how to handle constraint in the presence of these inconsistencies. If we recall the solution we used in the complete rule model, we argued that if two rules with opposite outcome apply to the same fact situation, we simply ignore them and leave the case unconstrained. In the other inconsistent scenario of a case violating a rule, we say that if a rule applies to a new fact situation, but the rule has already been violated by the decision of some existing precedent case, then we again ignore the rule.
Only in the scenario that all rules that apply have the same outcome do we obtain an obligation, only then is the courts decision forced. We can carry this idea into the new model, which leads us to the definition of a forcing rule. A forcing rule will be what we call a rule that is not part of any contradiction, and that is not violated by any case. These are the rules that can still lead to obligations of the court, while rules that are part of a contradiction, or that are violated by some case may lose their forcing power for some fact situations.

Definition 50 (Forcing lower/upper bound rule)**.** Given a case base Γ and a new fact situation F_{new}, a forcing lower bound rule is a rule $r_<$ belonging to a case

$$c = \langle F, r_<, r_>, (\omega, v)\rangle \in \Gamma$$

such that

1. $r_<$ applies to F_{new}, so
$$[r_<(d) \preceq_d X(d)$$
for all dimensions $d \in D$.
2. There is no upper bound rule $r'_>$ in Γ such that
   i) $r'_>$ applies to F_{new} and
   ii) The outcome of $r_<$ is higher than the outcome of $r'_>$, so
$$Outcome(r_<) < Outcome(r'_>)$$
3. There is no case
$$d = \langle F_d, rd_<, rd_>, (\omega, v_d)\rangle \in \Gamma$$
such that
   iii) r_v applies to F_d and
   iv) The outcome of d is lower than the outcome of $r_<$, so
$$v_d < Outcome(r_<)$$

The definition of a forcing upper bound rule is analogous.

It is important to note that the same rule can be a forcing rule or not, depending on the new fact situation that is presented. To show how to check whether a rule is forcing or not we give an example.

Example 27. We will use the case base we set up in Example 26, containing the three cases of Francesca, Gabriel and David. To recall, there was a rule conflict between the upper bound rule of Francesca's case and the lower bound rule of Gabriel's case, and the outcome of David's case violated the upper bound rule of Gabriel's case.
The first fact situation we consider is

$$X = \{(income, 680), (sqm, 55)\}$$

We can see that the upper bound rule of Francesca's case $rf_>$ applies, but is it forcing? The answer is no, because there is a lower bound rule $rg_<$ that also applies. Neither is therefore a forcing rule.
If we changed the fact situation however, to now be

$$X' = \{(income, 710), (sqm, 55)\}$$

we can see that the upper bound rule $rf_>$ no longer applies, while $rg_<$ still does. Since there is no other pair of conflicting rules, this means that $rg_<$ is now a forcing rule.

The second fact situation we consider is

$$Y = \{(income, 825), (sqm, 65)\}$$

We see that the upper bound rule of Gabriel's case $rg_>$ applies, but is it forcing? The answer is again no, because there exists a case, David's case, that violates $rg_>$. Therefore it is not forcing. The upper bound rule of David's case however is not violated by any case and it applies to Y, making it a forcing rule for this fact situation.

Using this concept of a forcing rule, we can then simply update the definition of constraint, following the intuition we outlined. We also need to update a fortiori constraint to only apply if it has not been violated before.

Definition 51 (Constraint of inconsistent case bases)**.** Given a case base Γ and a new fact situation F_{new}, the value v is a lower bound for the courts decision of F_{new}, if and only if

a) There exists a case $c = \langle F, r_<, r_>, (\omega, v)\rangle \in \Gamma$ such that for all dimensions, the values in F_{new} favors higher a value for ω than in F, and all cases $c' = \langle F', r'_<, r'_>, (\omega, v')\rangle \in \Gamma$ which favor higher values for ω than c in all dimensions have a higher outcome value.

or

b) There exists a forcing lower bound rule $r_<$ for a case $c \in \Gamma$ such that for all dimensions, the value in F_{new} favors higher values for ω than in $r_<$.

Formally, that means that v is a lower bound for ω if and only if

a) There exists a case $c = \langle F, r_<, r_>, (\omega, v)\rangle \in \Gamma$ such that

$$F(d) \preceq_d X(d)$$

for all dimensions and for all $c' = \langle F', r'_<, r'_>, (\omega, v')\rangle \in \Gamma$ with $F(d) \preceq_d F'(d)$ we have

$$v < v'$$

or

b) There exists a forcing lower bound rule $r_<$ for a case $c \in \Gamma$ such that

$$F(d) \preceq_d r_<(d)$$

for all dimensions.

The definition for v being an upper bound is analogous. The court must choose a value for ω that is greater than the greatest lower bound, and less than the least upper bound. The court is not bound in its decision of F_{new} if and only if there is neither a lower nor an upper bound.

To see that the definition matches the intuitions, we provide an example of how the generalized model constrains decisions.

Example 28. We want to check that the intuition of not allowing decisions that introduce new inconsistencies is expressed by the new definition of constraint. For this, we will consider an example and show a decision that is inconsistent, but since the inconsistency existed before it is still permitted. We use the same case base as in the previous examples, with the cases of Francesca, Gabriel and David. To recall, the fact situations were

$$F_f = \{(income, 650), (sqm, 60)\}$$
$$F_g = \{(income, 725), (sqm, 50)\}$$
$$F_d = \{(income, 800), (sqm, 70)\}$$

the rules were

$$rf_< = \{(income, 600), (sqm, 50)\} \to 15$$
$$rf_> = \{(income, 700), (sqm, 70)\} \to 25$$

for Francesca,

$$\begin{aligned} rg_< &= \{(income, 675), (sqm, 50)\} \to 30 \\ rg_> &= \{(income, 850), (sqm, 70)\} \to 35 \end{aligned}$$

for Gabriel and

$$\begin{aligned} rd_< &= rf_< \\ rd_> &= \{(income, 850), (sqm, 80)\} \to 40 \end{aligned}$$

for David. Finally, their full cases are

$$\begin{aligned} f &= \langle F_f, rf_<, rf_>, (\omega, 20)\rangle \\ g &= \langle F_f, rf_<, rf_>, (\omega, 35)\rangle \\ d &= \langle F_f, rf_<, rf_>, (\omega, 38)\rangle \end{aligned}$$

For the first example, consider the shelter being presented with the fact situation

$$X = \{(income, 680), (sqm, 55)\}$$

we saw before. We can obtain a forcing lower bound from rule $rf_<$ and a forcing upper bound from the a fortiori constraint of David's case. But are these the tightest forcing bounds? There is a tighter upper bound from rule $rf_>$ as well as a tighter lower bound from rule $rg_<$. However, these are part of an existing inconsistency and are therefore not forcing. Any decision will necessarily contradict one of the two rules, but that is acceptable as this inconsistency was already part of the case base. Therefore, any suitability score between 15 and 38 is a permitted outcome for this new fact situation. A decision for an outcome value outside this range however is prohibited, as it would violate a forcing rule, or go against the a fortiori constraint of a case that has not been violated. Such a decision would therefore be a new inconsistency.

This concludes the generalization of the new model we introduced. In summary, we have defined a model that uses dimensions for both the fact situation and the outcome, and that uses rules to justify the decisions. Furthermore, the generalization allows reasoning based on inconsistent case bases.
We now move on to the final contribution of this work.

3.4 Translating other reason models into BCL

The final contribution we make is a continuation of the work in [LLRS22]. Specifically, we present two translations of modifications of the reason model into the logic BCL [LL23]. Recall that we discussed BCL and the translation of the original reason model in Section 2.8.2. The motivation for these translations is the same as for the original translation of the reason model in [LLRS22]. Having a modal logic encoding allows us to define explanations of the decisions of the model. While we do not define these explanations in this thesis as it lies outside our scope, showing that other versions of the reason model can be translated into the same formal logic should be an encouraging result to continue this direction.

The two modifications we will provide translations for are

1. The factor-based reason model with inconsistent case bases
2. The complete rule model with finite dimensions

For both modifications we will provide the intuition of how we can encode its properties into BCL, formally define a translation of a case base into a formula, and show that the translation is correct. Before that, we will shortly recall the important ideas of BCL and the translation of the reason model. For the more detailed introduction, see Section 2.8.2.

The basic idea of BCL is to use a set of atomic propositions Atm_0 that represent the input features, and a special atomic proposition $t(x)$ that represents the classification or decision of that particular input. On the semantic side, this decision is done by a function, guaranteeing that each combination of input features will be decided for exactly one outcome.
Aside from that, the semantics uses a modal logic with classifier models, which contain states that represent the inputs and the decision function that maps a decision to each state.

The translation of the original reason model maps the input features to the factors, and the decision function to the decision of the court.
Precedential constraint is then encoded in two structural formulas, one which guarantees that each possible fact situation is covered by the model, and one which encodes the two-way monotonicity which forces the model to respect precedential constraint.
A case base is then translated into a BCL formula, to ensure that each classifier model that satisfies the formula decides the states corresponding to the precedent cases correctly.

The result for the original reason model is then, that a case base is consistent if and only if its translation is satisfiable by a classifier model which also satisfies the two structural formulas.

Having shortly recapped the basics of the BCL translation of the original reason model, we can now begin with the first modification.

3.4.1 Translating the factor-based reason model with inconsistent case bases

The first of the two modifications of the reason model that we translate into BCL is the generalization in [Canng] that allows for reasoning with inconsistent case bases. To recall exactly how the generalization works, see Section 2.4. In short, the generalization allows the initial case base to contain inconsistencies, and changes constraint to permit a decision as long as it does not introduce new inconsistencies to the case base.

In order to translate this model into BCL, we need to consider the two elements of the translation. The structural formulas that determine constraint, and the translation of the individual cases to encode specific case bases. For the structural part, we need to adjust the formulas `Compl` and `2Mon` to represent the new conditions for constraint.

The biggest challenge comes from the fact that the generalized reason model requires a differentiation between the precedent cases in the case base and any new fact situation. This differentiation is necessary, because the generalized reason model tolerates inconsistencies between the precedent cases, and only applies the two-way monotonicity to the new cases. Our encoding needs to reflect this difference. Were we not to introduce this tag, the two-way monotonicity would rule out all models with any inconsistency, because it has no information which inconsistencies we allow as they are part of the original case base. The translation would collapse into that of the original reason model. To allow for this tag, we extend the set of atomic propositions Atm by a new atom p with the intended meaning that whenever a state satisfies p, then it is a precedent, and when a state satisfies $\neg p$ then it is not a precedent.
From there on, we need to modify `Compl` to require each possible fact situations to be tagged as either a precedent or not, to avoid any models where the same case is both a precedent and not a precedent. The modification to `2Mon` adds a clause to the implication that reflects the different definition of when constraint applies in the generalized reason model.

For the formulas that encode the specific cases, we also need to make an adjustment, to tag the precedent cases and all states that are not precedents accordingly. With the ideas of the modifications clear, we now move on the the formal changes.

Changes to the structural formulas We will begin with the change to the structural formulas. Beginning with `Compl`, we need to make sure that not only do all models decide all possible fact situation, but additionally each possible fact situation must only have one state with one tag, either p or $\neg p$. This will make sure that in any model it is clear which states represent precedent decisions, and which states are subject to precedential constraint. The new formula is then

$$\texttt{Compl}^{inc} := \bigwedge_{X \subseteq Atm_0} \langle \emptyset \rangle\, cn_{X,Atm_0} \wedge (\langle \emptyset \rangle\, (cn_{X,Atm_0} \wedge p) \rightarrow [\emptyset]\, ((cn_{X,Atm_0} \rightarrow p)))$$

To better understand what we change for the formula `2Mon`, we need to recall one formulation of the conditions under which the court is forced to rule for one side over the other. In Section 2.4 we define constraint once based on the set of inconsistencies of a case base, this however will be difficult to capture in the language of BCL. Canavotto did provide us with a different way of formulating the constraint however, and that is what will allow us to understand the changed `2Mon` formula. In the alternative formulation, a decision of a new fact situation is forced for an outcome if there is some precedent case with the same outcome that supports the decision and there is no precedent case with opposite outcome which contradicts the decision. Phrasing this condition in terms of permission, if there are two precedent cases for opposite outcomes which would both normally force a decision for their respective outcome, then there is no constraint and both outcomes are permitted. For our formula this means that we need to add a condition to the implication that guarantees that the conclusion is only forced if there is no precedent case with the opposite outcome. Recall that we need to specifically tag the cases in the premise of the implication to be precedent cases. The resulting formula is then

$$\begin{aligned}\mathtt{2Mon}^{inc} := &\bigwedge_{s\in\{0,1\},X\subseteq Atm_0^s,Y\subseteq Atm_0^{\overline{s}}} (\langle\emptyset\rangle\,(cn_{X\cup Y,Atm_0}\wedge p\wedge t(s))\rightarrow\\ &\bigwedge_{Atm_0^s\supseteq X'\supseteq X,Y'\subseteq Y} (\neg\,\langle\emptyset\rangle\,(cn_{X'\cup Y',Atm_0}\wedge p\wedge t(\overline{s}))\rightarrow\\ &\qquad\bigwedge_{X'\supseteq X''\supseteq X,Y'\subseteq Y''\subseteq Y}[\emptyset]\,(cn_{X''\cup Y'',Atm_0}\wedge\neg p\rightarrow t(s))\Big)\Big)\end{aligned}$$

With this formula, we can then again define the class of classifier models that satisfy the theory of precedential constraint based on inconsistent case bases, which we will call $\mathbf{CM}^{prec,inc}$.

$$\mathbf{CM}^{prec,inc} := \Big\{C=(S,f)\in\mathrm{CM}\mid\forall a\in S: C,a\models\mathtt{Compl}^{inc}\wedge\mathtt{2Mon}^{inc}\Big\}$$

As before, satisfiability and validity for $\mathbf{CM}^{prec,inc}$ are defined as for CM.

Changes to the translation of cases In the second step, we need to change the translation function to tag all the cases that are contained in the case base of the generalized reason model we want to translate. Additionally, we need to ensure that all other cases will be tagged as new cases. The first part is accomplished in the translation function of individual cases.

Definition 52 (Translation of a case, generalized factor-based reason model)**.** We define a translation function tr^{inc} that maps a case $c=\langle F,r,s\rangle$ in the format of the factor-based reason model to a BCL formula. Recall that we can split the fact situation into the pro-plaintiff factors F^{π} and the pro-defendant factors F^{δ}.

$$tr^{inc}(\langle F,r,s\rangle) := \langle\emptyset\rangle\,(cn_{Premise(r)\cup F^{\overline{s}},Atm_0})\wedge p\wedge t(s)$$

This makes sure that all precedent cases are tagged as such, meaning that any inconsistencies among these cases will be tolerated. We will also make sure that all the sets of factors that are not part of the encoding of the precedent cases will be tagged correctly, by adding a clause to the extension of the function to entire case bases.

Definition 53 (Translation of a case base, generalized factor-based reason model)**.** Let Γ be a case base of the generalized factor-based reason model. We then define

$$tr^{inc}(\Gamma) := \bigwedge_{\langle F,r,s\rangle\in\Gamma} tr^{inc}(\langle F,r,s\rangle) \wedge \bigwedge_{X\in newFacts} \langle\emptyset\rangle\,(cn_{X,Atm_0} \wedge \neg p)$$

where $newFacts$ is the set of all sets of factors that are not part of the formula that encodes the precedent cases in Γ:

$$newFacts := \{X \subseteq Atm_0 \mid \nexists\,\langle F,r,s\rangle \in \Gamma\} \text{ s.t. } X = Premise(r) \cup F^{\bar{s}}$$

This definition of the translation function for cases and case bases will guarantee that the formula explicitly contains information about any state on whether it represents the decision of a precedent case or not. This then ensures the formula $\texttt{2Mon}^{inc}$ only applies monotonicity from precedent cases to new cases. To see exactly how this works, we consider an example.

Example 29. We consider the case base from Example 4, with the cases of Ursula and Victor. To recall, their cases were

$$u = \langle\{smallApp, prevPet, suffInc\}, \{prevPet\} \rightarrow \pi, \pi\rangle$$

for Ursula and

$$v = \langle\{prevPet, suffInc, shifts\}, \{shifts\} \rightarrow \delta, \delta\rangle$$

for Victor.

We will show how our translation would translate this case base, and then consider a formula that represents a new potential situation, to see how the formula $\texttt{2Mon}^{inc}$ applies precedential constraint. First however, here are the formulas that we get from translating the cases of Ursula and Victor:

$$\langle\emptyset\rangle\,(prevPet \wedge smallApp \wedge \neg suffInc \wedge \neg shifts \wedge p \wedge t(\pi))$$
$$\langle\emptyset\rangle\,(shifts \wedge prevPet \wedge suffInc \wedge \neg smallApp \wedge p \wedge t(\delta))$$

Note that since both cases are precedent cases, they both include the literal p, which means that a model that contains states that satisfy these formulas will also have states that can be used to satisfy the premise of the formula $\texttt{2Mon}^{inc}$. To see this in action, we now look at the case of William from Example 6, with the fact situation

$$F_w = \{shifts, smallApp, prevPet\}$$

To see whether the court is permitted to decide William's case for an outcome, we now need to create a formula that represents a potential decision, tag it with the literal $\neg p$ as a new case, and then see if it is affected by $\mathtt{2Mon}^{inc}$.
As we know from Example 6, the court is in fact not permitted to decide William's case for the plaintiff, and to see that this also holds for our translation, we only need to consider the one potential decision for the plaintiff that the court could make, which uses the rule $\{prevPet\} \to \pi$. This potential decision translates into the formula

$$\langle \emptyset \rangle \, (prevPet \wedge smallApp \wedge shifts \wedge \neg suffInc \wedge \neg p)$$

which requires that our model contains a state, that is not tagged as a precedent case, and that satisfies the premise of the rule as well as all factors for the opposite side, and does not satisfy the remaining factors, just like in the translation of the precedent cases.

Looking at these three formulas, we can now see that indeed, the formula $\mathtt{2Mon}^{inc}$ tells us that the state that satisfies

$$prevPet \wedge smallApp \wedge shifts \wedge \neg suffInc \wedge \neg p$$

must be decided for δ, and a decision for the plaintiff is therefore not permitted.
To see this, we just need to provide a witness for the first part of $\mathtt{2Mon}^{inc}$, that is a state that satisfies some subset of the atoms, satisfies p and is decided for δ. This state exists due to the translation of Victor's case, the subset of atoms in this instance is $X \cup Y$ where $X = \{shifts\}$ and $Y = \{prevPet, suffInc\}$.

For the second line, we need to argue that there is no state that satisfies p, is decided for π and that satisfies at least all the pro-defendant factors, and at most all the pro-plaintiff factors.
This is the case, as the only state that satisfies p and is decided for π does not satisfy $shifts$, so it does not satisfy all of the pro-defendant factors. To be precise to the formula, we have that there is no state that satisfies exactly the atoms in $X' \cup Y'$ where $X' = \{shifts, smallApp\}$ and $Y' = \{prevPet\}$.

Therefore, the third line tells us that any state that satisfies $\neg p$ and that satisfies exactly the atoms that are in $X'' \cup Y''$ where we have

$$X' = \{shifts, smallApp\} \supseteq X'' \supseteq \{shifts\} = X$$

and

$$Y' = \{prevPet\} \subseteq Y'' \subseteq \{prevPet, suffInc\} = Y$$

we must have that the decision for that state is δ. And as we can see, the state that satisfies our translation of William's potential decision does satisfy $shifts, smallApp$ and $prevPet$, as well as $\neg p$ so in order for $\mathtt{2Mon}^{inc}$ to hold, it must be decided for δ.

On the other hand, if we did the same with a case base that did contain some inconsistencies, we might get that there is a state that satisfies the second line of `2Mon`^{inc}, in which case the constraint might not apply. This of course is exactly the case where a new potential decision is inconsistent in a way that was already part of the case base.

A closer look at this translation reveals one final caveat. By statically tagging each case as either a precedent case or not, we necessarily have to rebuild this formula each time the case base is updated, so whenever a new case gets included. This is not a big issue, as we assume that an implementation of this translation would use a more sophisticated way of tagging formulas to allow for dynamically growing the set of precedent cases. For this work however, we are content with the theoretical result of having a translation that allows for obtaining information about a static case base by investigating the models of its translation. This brings us to directly to the final issue we need to address for this translation, how to formulate our results.

Results for the translation We clearly cannot state the same theorem as for the original reason model (Theorem 1), as it talks about consistent case bases. The entire point of the generalization is of course that the case base may be inconsistent. However, we can formulate a result in terms of precedential constraint, similar to the corollary for the original translation (see Corollary 1). The theorem for this translation therefore expresses that given an inconsistent case base and a new fact situation, the court is permitted to decide that case for some outcome if and only if there is a model for the formula that contains the case base and the new case with the decision for that outcome.

Theorem 4. *Let Γ be a (potentially inconsistent) case base of the generalized factor-based reason model and F the new fact situation. The court is permitted to decide F for outcome s using the rule r if and only if the formula*

$$tr^{inc}(\Gamma) \wedge \langle\emptyset\rangle\,(cn_{Premise(r)\cup F^{\bar{s}},Atm_0}) \wedge \neg p \wedge t(s)$$

is satisfiable in $\boldsymbol{CM}^{prec,inc}$.

Proof. Let Γ be a case base of the generalized factor-based reason model, and F_{new} be a new fact situation. We need to show two directions.

Direction "⇒" We first show the direction that if the court is permitted to decide F_{new} for some outcome s_{new} using rule r_{new} then the formula

$$tr^{inc}(\Gamma) \wedge \langle\emptyset\rangle\,(cn_{Premise(r)\cup F^{\bar{s}},Atm_0}) \wedge \neg p \wedge t(s)$$

is satisfiable in $\mathbf{CM}^{prec,inc}$.

Without loss of generality, assume then that the court is permitted to decide F_{new} for outcome π using rule r_{new}. This means, that including the case $\langle F_{new}, r_{new}, \pi\rangle$ in Γ

does not lead to any new inconsistency[6].

We define a classifier model $C = (S, f)$ and show that it is in the class $\mathbf{CM}^{prec,inc}$ and that it satisfies

$$tr^{inc}(\Gamma) \land \langle\emptyset\rangle\,(cn_{Premise(r)\cup F^{\delta},Atm_0}) \land \neg p \land t(\pi)$$

For this definition of $C = (S, f)$ we need some care in both the choice of a set of states as well as the definition of the decision function. Choosing these two parts correctly will then lead to a very easy argument as to why the classifier model we define does actually satisfy the formula.

Beginning with the set of states, let

$$\begin{aligned} S := \{X \cup \{p\} \mid X \in 2^{Atm_0} \text{ and } \exists\,\langle F, r, s\rangle \in \Gamma : X = Premise(r) \cup F^{\bar{s}}\}\cup \\ \{X \cup \{\neg p\} \mid X \in 2^{Atm_0} \text{ and } \nexists\,\langle F, r, s\rangle \in \Gamma : X = Premise(r) \cup F^{\bar{s}}\} \end{aligned}$$

This choice gives us all possible fact situations, and directly tags all states corresponding to precedent cases with p, and all other states with $\neg p$.

Defining the decision function f is more challenging. The idea of the definition is quite natural: For all the cases in the case base, we need to classify the corresponding states that are tagged as precedent states, i.e. the states that satisfy p, with the correct outcome that the case base tells us. For all the states tagged as new cases, so all states that satisfy $\neg p$ we need to classify them based on precedential constraint. The condition for precedential constrain is that of the generalized reason model, meaning that the decision is forced if there is some precedent case that supports it, and no precedent case that contradicts it.
Since the decision of the new fact situation is only permitted and not necessarily forced, we cannot guarantee that the state corresponding to that decision will be caught by these conditions, so we will explicitly define that f classifies it correctly.
Finally, for all other cases that are unconstrained, we simply choose ? and leave it undetermined.

Following this idea leads to an admittedly very complicated definition. For $a =$

[6] The proof of this theorem is very similar in some aspects to the proof of the result for the translation of the original reason model into BCL from [LLRS22]. However, the changes made for this translation complicate the formal details of the proof, to the point where a fully formal argument would be very hard to read. In the interest of still providing a solid argument for why our results are correct, we will make a quite detailed proof, but in some places that follow the argument of the original proof, we will omit some detail.

$Premise(r_{new}) \cup F^{\delta}_{new}$ let $f(a) = \pi$, and for all other $a \in S$ let

$$f(a) = \begin{cases} \pi & \text{if } C, a \models p \text{ and } \exists \langle F, r, \pi \rangle \in \Gamma \text{ s.t. } a = Premise(r) \cup F^{\delta} \\ & \text{or } C, a \models \neg p \text{ and } \exists \langle F, r, \pi \rangle \in \Gamma \text{ s.t.} \\ & a \cap Atm_0^{\pi} \supseteq Premise(r) \text{ and } a \cap Atm_0^{\delta} \subseteq F^{\delta} \\ & \text{and } \nexists \langle F', r', \delta \rangle \in \Gamma \text{ s.t. } a \cap Atm_0^{\delta} \supseteq Premise(r') \text{ and } a \cap Atm_0^{\pi} \subseteq F^{\pi} \\ \delta & \text{if } C, a \models p \text{ and } \exists \langle F, r, \delta \rangle \in \Gamma \text{ s.t. } a = Premise(r) \cup F^{\pi} \\ & \text{or } C, a \models \neg p \text{ and } \exists \langle F, r, \delta \rangle \in \Gamma \text{ s.t.} \\ & a \cap Atm_0^{\delta} \supseteq Premise(r) \text{ and } a \cap Atm_0^{\pi} \subseteq F^{\pi} \\ & \text{and } \nexists \langle F', r', \pi \rangle \in \Gamma \text{ s.t. } a \cap Atm_0^{\pi} \supseteq Premise(r') \text{ and } a \cap Atm_0^{\delta} \subseteq F^{\delta} \\ ? & \text{otherwise} \end{cases}$$

For this definition of our classifier model C we see right away that

$$C \models tr^{inc}(\Gamma)$$

since for every case $\langle F', r', s' \rangle$ in Γ by our definition of S we clearly have some $a \in S$ with

$$C, a \models p \text{ and } a = Premise(r') \cup F'^{\overline{s'}}$$

because we specifically tagged a state with p if there exists such a case. We also know that the decision function must decide every such a for the correct outcome, since that is a direct condition in the definition of f.
Furthermore, we have that for each subset of factors not part of the encoding of the precedent cases, there exists a state that corresponds to that subset and that satisfies $\neg p$, again because that is exactly what we used to define S.

We can also see that we must have

$$C \models \langle \emptyset \rangle \, (cn_{Premise(r) \cup F^{\delta}, Atm_0} \wedge \neg p \wedge t(\pi))$$

as the existence of some state a with

$$C, a \models cn_{Premise(r) \cup F^{\overline{s}}, Atm_0} \wedge \neg p$$

is guaranteed, again by our definition of S[7], and we specifically defined f to decide this state for π.

What remains is the more challenging part, arguing that C is in fact part of the class $\mathbf{CM}^{prec,inc}$. Showing that $C \models \texttt{Compl}^{inc}$ is again trivial due to our choice of S. Showing that $C \models \texttt{2Mon}^{inc}$ is the only part left.

[7] In the case that the new potential decision is identical to some existing decision, this does not hold. However, this case is not very interesting for precedential reasoning, so we do not include it our translation.

Suppose the opposite was true towards a contradiction. That means that there exists a state that is not a precedent state, that is constrained by the precedent states, but that has the opposite outcome than what precedential constraint required. Without loss of generality let then a be the constrained state, so a state with

$$C, a \models cn_{X'' \cup Y'', Atm_0} \wedge \neg p \wedge t(\pi)$$

let further b be some contradicting precedent, so a state with

$$C, b \models cn_{X \cup Y, Atm_0} \wedge p \wedge t(\delta)$$

and we know that there cannot be any state c that is a supporting precedent with

$$C, c \models cn_{X' \cup Y', Atm_0} \wedge p \wedge t(\pi)$$

where $X \subseteq X'' \subseteq X' \subseteq Atm_0^{\delta}$ and $Atm_0^{\pi} \supseteq Y \supseteq Y'' \supseteq Y'$.
This setup would be such a counterexample to $\texttt{2Mon}^{inc}$, and would therefore have to be present for our assumption that $\texttt{2Mon}^{inc}$ is not satisfied by our model.

There are two options for state a. It is either the state that corresponds to the new potential decision, so we have $a = Premise(r_{new}) \cup F_{new}^{\delta}$ or it is any of the other sets of factors that do not correspond to any of the precedent cases.

If a is just one of the sets of factors that do not correspond to any of the precedent cases, then our definition of f leads to an immediate contradiction. The facts that there exists a state b as described and that no state c as described exists imply that the case base must contain a case that corresponds to b and no case that would correspond to c.
This however is exactly the condition for f to decide a for δ. Since f is a function, it can only decide a for one of the outcomes, so we get a contradiction.

If a corresponds to the new potential decision, so $a = Premise(r_{new}) \cup F_{new}^{\delta}$, then assuming that there is a state b as described and no state c as described leads to a contradiction to the assumption that the potential decision of the new case was permitted, since again the assumptions about the states imply the existence of a supporting precedent for the outcome δ and no contradicting precedent. This would mean that the decision for π would not be permitted.

In conclusion, we get a contradiction to our assumption that $\texttt{2Mon}^{inc}$ is not satisfied, which means that indeed C is a classifier model of the class $\mathbf{CM}^{prec,inc}$ and it satisfies

$$tr^{inc}(\Gamma) \wedge \langle\emptyset\rangle \left(cn_{Premise(r) \cup F^{\overline{s}}, Atm_0}\right) \wedge \neg p \wedge t(s)$$

Direction "⇐" What remains is the direction that if the formula

$$tr^{inc}(\Gamma) \wedge \langle\emptyset\rangle \left(cn_{Premise(r) \cup F^{\overline{s}}, Atm_0}\right) \wedge \neg p \wedge t(s)$$

is satisfiable in $\mathbf{CM}^{prec,inc}$ then the potential decision of fact situation F for outcome s using rule r is permitted.
We show this by contraposition, showing instead that if the decision is not permitted, then the formula is unsatisfiable.

Without loss of generality, assume then that deciding the new fact situation F for outcome π using rule r is not permitted. The assumption that the potential decision is not permitted is equivalent to a potential decision for δ being forced. Or, in the other formulation of the generalized reason model there is a supporting precedent for deciding the case for δ and there is no contradicting precedent for deciding the case for δ.

To show then that the formula

$$tr^{inc}(\Gamma) \wedge \langle\emptyset\rangle \left(cn_{Premise(r)\cup F^{\delta,Atm_0}}\right) \wedge \neg p \wedge t(\pi)$$

is unsatisfiable in $\mathbf{CM}^{prec,inc}$, we assume towards a contradiction that there was some classifier model $C = (S, f)$ from the class $\mathbf{CM}^{prec,inc}$ that satisfied it.

From the assumption that the decision for δ is forced, and that there must therefore be some supporting precedent but no contradicting precedent, we get that there must be a case $c = \langle F_{supp}, r_{supp}, \delta\rangle \in \Gamma$ such that $Premise(r_{supp}) \subseteq F^{\delta}$ and $F^{\pi} \subseteq F^{\pi}_{supp}$, which is a supporting precedent. At the same time, there cannot be any case $d = \langle F_{contr}, r_{contr}, \pi\rangle \in \Gamma$ with $Premise(r_{contr}) \subseteq F^{\pi}$ and $F^{\delta} \subseteq F^{\delta}_{contr}$ which would be a contradicting precedent.

Laying these subset relations out, we get the following:

$$Premise(r_{supp}) \subseteq F^{\delta} \subseteq F^{\delta}_{contr}$$

for the sets of pro-defendant factors and

$$Premise(r_{contr}) \subseteq F^{\pi} \subseteq F^{\pi}_{supp}$$

for the sets of pro-plaintiff factors.
Since we assume that our model satisfies the translation of the case base, we also know that all the states corresponding to precedent cases are tagged with p, while the potential decision is tagged as $\neg p$.

Laying this over the formula $\texttt{2Mon}^{inc}$ however reveals, that for the implication to be satisfied, the state that corresponds to the potential decision must be decided for δ. This is a contradiction to the functionality of f however, since it cannot decide that state for π to satisfy

$$tr^{inc}(\Gamma) \wedge \langle\emptyset\rangle \left(cn_{Premise(r)\cup F^{\overline{s}},Atm_0}\right) \wedge \neg p \wedge t(\pi)$$

and simultaneously for δ to satisfy $\texttt{2Mon}^{inc}$.

In conclusion, our assumption that there the formula

$$tr^{inc}(\Gamma) \wedge \langle \emptyset \rangle \left(cn_{Premise(r) \cup F^{\overline{s}}, Atm_0} \right) \wedge \neg p \wedge t(s)$$

is satisfiable in class $\mathbf{CM}^{prec,inc}$ leads to a contradiction, and therefore the formula must be unsatisfiable. □

We now move on to the second modification of the reason model we encode into BCL, this modification being the complete rule model.

3.4.2 Translating the complete rule model into BCL

The idea of encoding any of the dimensional reason models into BCL, a binary logic might seem strange. And indeed, translating these models with their full expressiveness is not something we aim for in this work. However, with the very reasonable restriction of requiring each dimension to be finite, we are able to easily translate the many-valued dimensions into binary propositional atoms.
We will first motivate why this restriction is indeed unproblematic, before presenting our approach to encoding dimensions and the complete rule model into BCL. We will again show the necessary changes to the structural formulas and the translation of cases and case bases.

First however, we want to argue why restricting dimensions to be finite is not actually a big restriction. The simple answer is that there will be few, if any scenarios where having infinitely many values is going to be necessary. We would even claim that very few scenarios even need a substantially large number of values. In the legal setting, it is hard to imagine that the court will consider cases where some fact would have to be represented by one of infinitely many values, neither for magnitude nor for precision. Some easy examples would be age of the plaintiff, clearly it is not necessary to consider plaintiffs that are over 200 years old, so having a maximum value for age is reasonable. At the same time, it does not matter whether the plaintiff is 20 years, 3 months and 4 days old or 20 years, 3 months and 10 days, so we do not need infinitely many values for precision either.
We claim that for most applications, restricting the range of values that the dimensions can take to some finite set is possible with no actual loss of necessary expressive power.

Restricting our considerations to finite dimensions will then allow us to use an easy encoding of the values into our boolean logic. With the encoding in place, all we need is a formula to express the relation of two values, to encode the dimensional version of the two-way monotonicity. The remaining changes to the formulas will come quite naturally from the encoding of the dimensions.

Let us therefore begin with our encoding of dimensions. The encoding we used for factors was naturally just a simple propositional atom, as both are binary concepts. A finite dimension however can take one of many values. Therefore, our set of atomic propositions needs to change. We now need to include a set of atoms for each dimension which is used to encode the values of that dimension. This encoding is then simply a binary encoding of the value. All we need is some assumptions on the ordering of the values of a dimension, and we can easily map each dimension onto some range of numbers 0-k, and then use propositional atoms to encode these values.

Let then $D = \{d_1, d_2, \dots, d_n\}$ be a domain containing finite dimensions. We first need to map each dimension to an interval of natural numbers. The mapping relies on the ordering of the values of the dimension.
As before, we will assume that each dimension is ordered in a way where for two values with $p \preceq_d q$ we get that p is stronger for the defendant, and q is stronger for the plaintiff. We then encode the values of a dimension by mapping the minimal value with respect to $\preceq_d$ to 0, the next smallest value to 1 and so on, until we map the maximal value with respect to $\preceq_d$ to $k-1$, the number of values in the dimension.

What is left is to map these numbers into boolean formulas. This however is very simple, as we just need to introduce enough atomic propositions to do a binary encoding of the number. Specifically, for a dimension with k values that means we need $\log_2(k)+1$ atoms. For example, to represent the value 13 of some dimension, we would use the formula $c^4 \wedge c^3 \wedge \neg c^2 \wedge c^1$ which is satisfied by an interpretation that corresponds to the binary number 1101 which is 13. Doing this for each dimension yields the new set of features. For our domain D with dimensions d_1 to d_n, where each dimension d_i has k_i values, we get the following set of atomic propositions to represent the input, which is still called Atm_0.

$$Atm_0 = \bigcup_{d_i \in D} \left\{ c^1_{d_i}, c^2_{d_i}, \dots, c^{k_i}_{d_i} \right\}$$

It is important to note, that due to this binary encoding, there may be values that are encoded by some formula, but that are not actually part of the dimension. This is however not an issue, as we can simply ignore any model that assigns the atoms of some dimension to a value outside of the actual range. The additional values do not interfere with any reasoning.
As we are still dealing with a binary outcome, the atomic formulas for the decision function are unchanged. To get a clearer idea of how this representation of dimensions work, we will look at a small example.

Example 30. To get an idea of how we encode the dimensions, we will use the same dimensions we introduced in Example 8, which were

$$\begin{array}{ccc} income & : & [0,\infty) \\ sqm & : & [0,\infty) \\ pet & : & \{none, fish, cat, dog\} \\ work & : & \{shifts, regular, home\} \\ roommate & : & \{no, yes\} \end{array}$$

Now as we mentioned, we will only consider finite dimensions, which means we have to restrict the values somewhat. Instead of using the interval $[0, \infty)$ for the income and the square meters, we will use the interval $[0, 511]$ for income, and the interval $[0, 255]$ for square meters.
We will interpret the value of the income dimension to be the actual income divided by 10, sacrificing some precision but reducing the number of atoms we need to encode it. For square meters, we will just use the value to be the actual square meters.

For the other dimensions, we just need to map each value to a number, taking into account that we want 0 to stand for the value most favorable for the defendant. That means that for the dimension *pet* we map *none* to 0, $fish$ to 1, *cat* to 2 and *dog* to 3.
Following the same procedure for the remaining two dimensions leaves us with the following numerical, properly ordered and finite dimensions:

$$\begin{array}{ccc} income & : & [0, 511] \\ sqm & : & [0, 255] \\ pet & : & \{0, 1, 2, 3\} \\ work & : & \{0, 1, 2\} \\ roommate & : & \{0, 1\} \end{array}$$

To encode these dimensions into binary, we then introduce a set of new atoms for each dimension. We can notice, that our choice of interval for *income* and *sqm* means that we can use 9 and 8 atoms respectively, without encoding any values that are not part of the interval. This is not the case for the dimension *work*, where we have 3 values, meaning we need 2 atoms to encode it, but that means that we actually have 4 values that we can express. This is not an issue however, we just ignore any assignment that represents the value 4 for the dimension *work*.

The atoms for *income* would then be

$$\{c^1_{income}, c^2_{income}, \ldots c^9_{income}\}$$

and they represent a simple binary encoding, where c^1_{income} represents the most value bit.

If we consider a state of our classifier model, if the state contains the atoms

$$\{c^2_{income}, c^4_{income}, c^5_{income}, c^7_{income}, c^8_{income}\}$$

then that state represents the binary number 010110110, which is the value 182, and since we scale this value by a factor of 10, we get that this state represents an income value of 1820.

The same can then be applied to all other dimensions, which results in each state containing a value assignment for each dimension.

Having introduced our representation of dimensions, we now begin by looking how we need to modify the two structural formulas `Compl` and `2Mon`, before we look at the translation of the individual cases and the case base as a whole.

Changes to the structural formulas Looking at the formula `Compl`, one might assume that since we change the representation of our cases we need to adapt the formula to work for this new representation. Recall that for the factor-based reason models, `Compl` was used to express that each possible configuration of present and absent factors must have a corresponding state in the model. For the dimensional setting that would mean that for each possible value assignment there must be a corresponding state.
However, due to our encoding of dimensions, we actually do not need to change the formula at all, since each subset of our atomic propositions will correspond to exactly one value assignment. Therefore, requiring a state for each subset of the atomic propositions will achieve the desired result without any change needed. As we pointed out above, the binary encoding might end up with more encoded values than the dimension actually has, but as we can simply ignore those states when looking at our models, we do not need to exclude those meaningless states.

While we do not need to adjust the formula `Compl`, we do need to adjust the formula `2Mon`. The change however is again not too complicated, as we simply need to adjust the condition that encodes the monotonicity of the classification function.
Since this condition is quite simple for the complete rule model, stating it in BCL ends up being quite straightforward. All we need to encode is that if there exists some precedent case with some assignment and that is decided for outcome s, then all states with an encoding that represents values stronger for s than the ones in the precedent case must also be decided for s. The actual formula turns out to be quite technical and complex, since we need to split the atoms of any state into the sets of atoms that encode one dimension, and compare two binary encodings, however it captures exactly this intuition. We will still provide the formula so that we can reason with it in the proof.

To find the states where the values are stronger than in the precedent, we need to compare the two binary numbers that represent the value for each dimension. We do not provide a formula that compares two binary encodings, as it is quite technical. We know that such a formula exists from the fact that there is a hardware circuit called magnitude comparator which compares two binary numbers (for more information see [Wak18]).
In this work, we will write the shorthand

$$\texttt{LEQ}((c_1^1, c_1^2, \ldots, c_1^k), (c_2^1, c_2^2, , \ldots, c_2^j))$$

for the formula that is true if and only if the number represented by the first tuple of atoms is less or equal to the number represented by the second tuple of atoms, and

$$\texttt{GEQ}((c_1^1, c_1^2, \ldots, c_1^k), (c_2^1, c_2^2, , \ldots, c_2^j))$$

for the formula that is true if and only if the number represented by the first tuple is greater or equal to the number represented by the second tuple of atoms.

Aside from these shorthands, we need one additional notation to make the formula more readable. For any subset $X \subseteq Atm_0$ we will write $d_i(X)$ to refer to the tuple of those atoms in X that are used to encode the value of d_i. For example, if X contains the atoms c_1^1, c_1^3 and c_1^4 and dimension d_1 is encoded with 4 atoms, then we have $d_1(X) = (c_1^1, \neg c_1^2, c_1^3, c_1^4)$ which we can pass to the `LEQ` or `GEQ` function to get a formula that compares the encoded value of d_i to the one encoded by another subset.

For two subsets $X, Y \subseteq Atm_0$ the statement $\texttt{QEQ}(d(X), d(Y))$ then gives us a formula that is true if and only if X encodes a value assignment that assigns d a higher value than the assignment encoded in Y.

Since we need to check for greater or equal values if the precedent was decided for the plaintiff and less or equal values if the precedent was decided for the defendant we need to separate those two cases. However, the basic structure of the formula remains, where we fix some subset X and if there is a state where X holds, then we force all states that represent cases with stronger (or weaker) fact situations to be decided in the same way as the state where X holds.

$$\begin{aligned}
\texttt{2Mon}^{cr} := \bigwedge_{X \subseteq Atm_o} & (\langle\emptyset\rangle\, (cn_{X,Atm_0} \wedge t(\pi)) \rightarrow \\
& \bigwedge_{Y \subseteq Atm_0} \left([\emptyset] \left(cn_{Y,Atm_0} \wedge \left(\bigwedge_{d_i \in D} \texttt{LEQ}(d_i(X), d_i(Y)) \right) \rightarrow t(\pi) \right)\right)\Bigg) \wedge \\
& \qquad (\langle\emptyset\rangle\, (cn_{X,Atm_0} \wedge t(\delta)) \rightarrow \\
& \qquad \bigwedge_{Y \subseteq Atm_0} \left([\emptyset] \left(cn_{Y,Atm_0} \wedge \left(\bigwedge_{d_i \in D} \texttt{GEQ}(d_i(X), d_i(Y)) \right) \rightarrow t(\delta) \right)\right)\Bigg)
\end{aligned}$$

With the two structural formulas in place, we once again define a new class of classifier models $\mathbf{CM}^{prec,cr}$ defined by the set

$$\mathbf{CM}^{prec,cr} := \{C = (S, f) \in \mathrm{CM} \mid \forall s \in S : C, s \models \texttt{Compl} \wedge \texttt{2Mon}^{cr}\}$$

as the class of classifier models that satisfy the theory of precedential constraint of the complete rule model.

Changes to the translation of cases The translation of cases and case bases turns out to be quite simple, and is mostly a result of the encoding of dimensions. Since each case only contains the rule that is relevant for constraint and the outcome, we simply follow the same idea as for the translation of the original factor model. All we need to add is some notation to represent the encoding of the values. We will write $bin(n)$ to map any natural number n to the formula that represents the binary encoding. We of course also assume our domain D to contain only finite dimensions.

Definition 54 (Translation of a case, complete rule model)**.** We define a translation function tr^{cr} that maps a case $c = \langle F, r, s\rangle$ in the format of the complete rule model with finite dimensions to a BCL formula.

$$tr^{cr}(\langle F, r, s\rangle) := \langle\emptyset\rangle \left(\left(\bigwedge_{d\in D} bin(r(d))\right) \wedge t(s)\right)$$

And naturally the extension for case bases.

Definition 55 (Translation of a case base, complete rule model)**.** Let Γ be a case base of the complete rule model with finite dimensions. We then define

$$tr^{cr}(\Gamma) := \bigwedge_{\langle F,r,s\rangle\in\Gamma} tr^{cr}(\langle F, r, s\rangle)$$

To see how this translation works, we look at an example.

Example 31. We translate the case base of Example 15 with the cases of Maria and Linus into a BCL formula, and show some parts of the structural formula $\texttt{2Mon}^{cr}$.
To start, we need to change the domain from the one in Example 15 to the one from Example 30, which we can then encode as we did in the previous example.
Since none of the values in the cases of Maria and Linus are outside the new dimensions, we do not need to change anything in their cases. Recall that we need to encode the values of the rules of the cases, which means we need to encode the rule of Maria' case

$$r_m = \{(sqm, 80), (pet, cat), (income, 1000), (work, regular)\} \to \delta$$

and the rule

$$r_l = \{(sqm, 90), (pet, cat), (income, 1000), (work, home)\} \to \pi$$

for the case of Linus. From these rules we also see that the court ruled in favor of the defendant in the case of Maria, and in favor of the plaintiff in the case of Linus.

Using the encoding of our dimensions and the translation function we can obtain the following formula for the case of Maria:

$$tr^{cr}(m) = \langle\emptyset\rangle \Big(\neg c^1_{sqm} \wedge c^2_{sqm} \wedge \neg c^3_{sqm} \wedge c^4_{sqm} \wedge \neg c^5_{sqm} \wedge \neg c^6_{sqm} \wedge \neg c^7_{sqm} \wedge \neg c^8_{sqm} \wedge$$
$$\neg c^1_{income} \wedge \neg c^2_{income} \wedge c^3_{income} \wedge c^4_{income} \wedge \neg c^5_{income} \wedge \neg c^6_{income} \wedge c^7_{income} \wedge \neg c^8_{income} \wedge \neg c^9_{income} \wedge$$
$$c^1_{pet} \wedge \neg c^2_{pet} \wedge \neg c^1_{work} \wedge c^1_{work} \wedge t(\delta)\Big)$$

which encodes the values of her rule using our binary encoding and puts the encoding in conjunction with a decision for the defendant. For the case of Linus we get the formula

$$tr^{cr}(l) = \langle\emptyset\rangle \Big(\neg c^1_{sqm} \wedge c^2_{sqm} \wedge \neg c^3_{sqm} \wedge c^4_{sqm} \wedge c^5_{sqm} \wedge \neg c^6_{sqm} \wedge c^7_{sqm} \wedge \neg c^8_{sqm} \wedge$$
$$\neg c^1_{income} \wedge \neg c^2_{income} \wedge c^3_{income} \wedge c^4_{income} \wedge \neg c^5_{income} \wedge \neg c^6_{income} \wedge c^7_{income} \wedge \neg c^8_{income} \wedge \neg c^9_{income} \wedge$$
$$c^1_{pet} \wedge \neg c^2_{pet} \wedge c^1_{work} \wedge \neg c^1_{work} \wedge t(\pi)\Big)$$

We can then look at one part of the $\texttt{2Mon}^{cr}$ formula and see how it derives decisions based on precedential constraint.
As the representation of our facts requires a lot of literals however, we will instead just write the binary string our encoding represents for readability.

The formula $\texttt{2Mon}^{cr}$ then contains the clause

$$\begin{aligned} &\langle\emptyset\rangle\,(0101000_{sqm} \wedge 001100100_{income} \wedge 01_{pet} \wedge 10_{work} \wedge t(\delta)) \rightarrow \\ &\qquad [\emptyset]\,((01001111_{sqm} \wedge 001100100_{income} \wedge 00_{pet} \wedge 10_{work} \wedge \\ &\qquad \texttt{GEQ}(0101000, 01001111) \wedge \texttt{GEQ}(001100100, 001100100) \wedge \\ &\qquad\qquad \texttt{GEQ}(01, 00) \wedge \texttt{GEQ}(10, 10)) \rightarrow t(\delta)) \end{aligned}$$

which tells us, that if there is a state that satisfies the encoded values and that is decided for the defendant, then for any state that satisfies the second encoding, and where the values encoded by the first state are greater or equal the those encoded by the second state, that new state must also be decided for the defendant.
We can see how the formula captures our idea of the constraint of the complete rule model very directly.

Finally, we can move on to the results for this translation.

Results for the translation Since we do have a definition for consistency (see Definition 47 in Section 3.3.1), we can actually state a theorem for case base consistency, and the corollary for precedential constraint.

Theorem 5. *Let Γ be a case base of the complete rule model with finite dimensions. Then Γ is consistent if and only if the formula $tr^{cr}(\Gamma)$ is satisfiable in the class $\boldsymbol{CM}^{prec,cr}$.*

Corollary 2. *Let Γ be a consistent case base of the complete rule model with finite dimensions and F be a new fact situation. The court is permitted to decide F for outcome s using the rule r if and only if the formula*

$$tr^{cr}(\Gamma) \wedge tr^{cr}(\langle F, r, s\rangle)$$

is satisfiable in $\boldsymbol{CM}^{prec,cr}$.

We only prove the theorem, as the corollary follows right away.

Proof. Let Γ be a case base of the complete rule model with finite dimensions. Let D be the domain. This proof follows the structure of the proof of the original theorem from [LLRS22] even more closely, as we prove the exact same equivalence. The only thing that differs is the definition of consistency.
We will begin as before with the direction that if the Γ is consistent, then the formula obtained by the translation is satisfiable.

Direction "⇒" Assume that Γ is consistent. We construct a classifier model and show that it satisfies the formula $tr^{cr}(\Gamma)$. We assume the encoding of all dimensions in D as we have described it above.

To define the classifier model, we again define a set of states and a decision function. The set of states is even simpler as in the case of the generalized reason model, as our dimension encoding guarantees that by having a state for each subset of our set of atomic propositions, we cover all possible value assignments to all dimensions.
For the decision function, we just need to make sure that any state that corresponds to an assignment which is constrained by some rule in the case base is decided in accordance with the precedential constraint. For this, we just need to compare the values encoded in the state with the values of the rules in the case base[8]. Let therefore $C = (S, f)$ be a classifier model with $S = 2^{Atm_0}$ and for $a \in S$ let

$$f(a) = \begin{cases} \pi & \text{if } \exists \langle F, r, \pi\rangle \in \Gamma \text{ s.t. } r(d) \leq d(a) \text{ for all } d \in D \\ \delta & \text{if } \exists \langle F, r, \delta\rangle \in \Gamma \text{ s.t. } r(d) \geq d(a) \text{ for all } d \in D \\ ? & \text{otherwise} \end{cases}$$

be the decision function. All that is left is to verify that this classifier model is in the class $\mathbf{CM}^{prec,cr}$ and that it does indeed satisfy $tr^{cr}(\Gamma)$.

Checking that it satisfies $tr^{cr}(\Gamma)$ is trivial, as there is a state for every assignment, and naturally for all cases in Γ we will have that

$$\exists \langle F, r, s\rangle \in \Gamma \text{ s.t. } d(r) = d(a) \text{ for all } d \in D$$

is true for some state a, and therefore the decision function will decide the state accordingly.

Checking that $C \models \texttt{Compl}$ is also trivial due to our choice of S.

What remains is showing that $C \models \texttt{2Mon}^{cr}$.
Towards a contradiction, assume the opposite, that C does not satisfy $\texttt{2Mon}^{cr}$. Without loss of generality, assume that there is a state a which satisfies $cn_{X,Atm_0} \wedge t(\pi)$ for some $X \subseteq Atm_0$, and some other state b which satisfies

$$cn_{Y,Atm_0} \wedge \left(\bigwedge_{d_i \in D} \texttt{LEQ}(d_i(X), d_i(Y)) \right) \wedge t(\delta)$$

[8] We note that this need for comparing the encoded value of the subset against the high level value of the case leads to an imprecision in notation, as we are using $d(a)$ to get the atoms that encode dimension d in state a, and also $r(d)$ to get the value that the rule r assigns to dimension d. We therefore cannot actually directly compare the two, but it should be fairly intuitive to understand the meaning of the expression regardless.

where $Y \subseteq Atm_0$.
By our definition of f, this means that there must be cases $\langle F, r, \pi\rangle$ and $\langle F', r', \delta\rangle$ in Γ such that $d(r) \leq d(a)$ and $d(r') \geq d(b)$ for all $d \in D$. From our assumption on b we also get that $d(a) \leq d(b)$ for all $d \in D$.
This however contradicts the assumption that Γ was consistent, as by transitivity we get that $d(r) \leq d(r')$ but the outcome of r is π, and the outcome of r' is δ.

Direction "$\Leftarrow$" We again show this direction by contraposition, meaning we show that if Γ is inconsistent, then the formula $tr^{cr}(\Gamma)$ is unsatisfiable in $\mathbf{CM}^{prec,cr}$.

Assume then that Γ is inconsistent. By the definition of inconsistency, that means that there are two cases $c, d \in \Gamma$ with $c = \langle F, r, \pi\rangle$ and $d = \langle F', r', \delta\rangle$, and where $r(d) \leq r'(d)$ for all $d \in D$.

We show that any classifier model that satisfies $tr^{cr}(\Gamma) \wedge \texttt{Compl}$ cannot satisfy $\texttt{2Mon}^{cr}$. Towards a contradiction, assume $C = (S, f)$ is a classifier model that satisfies $tr^{cr}(\Gamma) \wedge \texttt{Compl}$ as well as $\texttt{2Mon}^{cr}$. From the cases c and d and the assumption $C \models tr^{cr}(\Gamma)$ we get that there must be a state $a \in S$ such that a encodes the value assignment of the rule r and $C, a \models t(\pi)$. Furthermore, there must be a state $b \in S$ such that b encodes the value assignment of r'. This means that there are two subsets $X, Y \subseteq Atm_0$ such that $C, a \models cn_{X,Atm_0}$ and $C, b \models cn_{Y,Atm_0}$. Furthermore, since we have $r(d) \leq r'(d)$ for all $d \in D$, we get that

$$C, b \models cn_{Y,Atm_0} \wedge \left(\bigwedge_{d_i \in D} \texttt{LEQ}(d_i(X), d_i(Y)) \right)$$

must hold. That means, that for $\texttt{2Mon}$ to hold we must get that $C, b \models t(\pi)$. This however contradicts the functionality of the decision function, as it cannot decide b for both δ and π. Therefore, there cannot be a model that satisfies $tr^{cr}(\Gamma) \wedge \texttt{Compl}$ and $\texttt{2Mon}^{cr}$ at the same time, and the formula is therefore unsatisfiable. □

This concludes the contributions of this thesis, which include the definition of a new model of precedential reasoning, the generalization of this new model to enable reasoning based on inconsistent case bases, and the translation of two reason models for precedential reasoning into the logic BCL.

CHAPTER 4

Related work

As all the previous models we presented stem from the legal domain and are motivated by legal reasoning, the related works are also primarily from legal contexts. For this work and specifically the longer term goal of developing a hybrid approach for normative knowledge acquisition we outlined in Chapter 1, the considerations that are specifically about the *legal* aspects of the models are not always relevant. We will therefore focus on the formal aspects and basic ideas of other models, without spending too much time on the legal principles motivating them.
Another important remark is that we do not go into much detail regarding the topic of automating the concrete acquisition of case information at much length. While this is clearly an integral aspect of making the model we discussed (or more importantly the models that will be developed in the future) useful for AI, the focus of this thesis is on the knowledge representation and reasoning.

In this chapter, we will focus instead on additional perspectives or aspects of precedential reasoning models in the literature, as well as other developments towards using these models in AI applications.
There are four aspects that of particular interest to us.

Dimensions. We have presented two perspectives on precedential reasoning with dimensions in detail in Chapter 2, and want to shortly look at two other ways to handle dimensions in Section 4.1.

Hierarchies. We have presented a single-step version of a hierarchical approach as inspiration for the way to obtain a many-valued outcome, but in the literature there is a multitude of concepts and models that use more structured fact representations. We will look at some of this work in Section 4.2. Further developments of the new model we presented in Chapter 3 might also include using a hierarchical structure, so taking a look at different existing approaches is valuable.

Missing case information. We have so far ignored the issue of incomplete information when discussing the previous models, both for the sake of simplicity, and since those previous models also ignored it. For practical use however, it plays an important role, and will certainly be part of future work on these models. Both for the factor- and dimension-based scenario there have been proposals on how to deal with absent factors, or absent values for dimensions. In Section 4.3 we will discuss some ideas that relate to handling incomplete information, both from a conceptual as well as a practical standpoint.

Explanations. The subject we discuss in Section 4.4 will touch more on the connection of the reasoning models to formal logic. Specifically, we will take a look at one of the uses of the logical representation of the reason model in the form of explanations. In AI, the subject of explanations plays a big part, as being able to explain the systems decisions increases the transparency and therefore the trust in the system. Future work on models like the one we developed will very likely also be on providing such explanations for its decisions.

4.1 Other views on dimensions

The use of dimensions to represent facts has been part of precedential reasoning since the systems of HYPO [RA87] and CATO [AA97] were introduced, which may be seen as the starting point of automated precedential reasoning. The two models we have presented provide two possible ways of using dimensions to represent cases with rules, and two mechanisms to include dimensions in the reasoning process. Horty reduces the dimensional representation to factors with reference values [Hor19, Hor20], while Prakken provides reference values using the same assignments that are used for the fact situation. We will now present two more approaches to dimensions, starting with the model of Rigoni [Rig18]. He also reduces dimensions to factors, however his model does not rely on factors that have a reference value directly attached. Instead, his model splits each dimension into a pro plaintiff and a pro defendant side, with the value (or region of values) between being called the *switching point*. If a fact situation then assigns some dimension a value on the pro plaintiff side of the switching point, then the pro plaintiff factor for that dimension is present. This avoids the issue of Horty's model, where for any given value p there might be a magnitude factor for each side that the court uses, which leads to counter-intuitive situations (see Example 13). In Rigoni's model, each value of a dimension can only be used to have one factor be present, which then always favors the same side. One prominent point of critique for Rigoni's model is that identifying a switching point (if such a point even exists), is a nontrivial and potentially subjective task, which makes it difficult to apply in practice.
The second approach we want to discuss is that of Bench-Capon and Atkinson [BA17]. Their works relate to a framework called an Abstract Dialectical Framework (ADF),

which is used to represent domain knowledge [AAB16]. This representation uses both factors and dimensions, and is hierarchical in nature. Starting from a set of basic nodes, each higher level node's value is determined only by the values of the input. Importantly, a previous result [AAB15] is that every ADF can be rewritten so that the structure is essentially a binary tree. This allows for a full distinction of all possible input nodes for any given higher level node. For example, if both input nodes are factors, then the condition for the higher level node would be a boolean function of the two inputs. This structure and the different cases for determining the values of higher level nodes differs significantly from the approaches we have seen so far, and includes a lot of consideration in how to formulate the conditions, especially for cases like both inputs being dimensions. For this case, they use a form of geometric reasoning, with precedents partitioning the two-dimensional space into sections that favor a certain outcome. This form of specifically adapted reasoning is enabled by restricting each step to only rely on two inputs, which contrasts the decisions in the previous models, which are usually unbound in the number of input dimensions.

Staying with the approach of Bench-Capon and Atkinson, we will now discuss another aspect of their framework that we mentioned already, the hierarchical structure.

4.2 Other hierarchical approaches

The ADF representation of domain knowledge used in the works of Bench-Capon and Atkinson is a hierarchical one. Structuring domain knowledge and aspects of a case in a hierarchy, where abstract concepts depend on base-level concepts has been part of legal case-based reasoning since the start. While some of the motivation for this comes from the way legal cases are structured, another reason that is more relevant to the generalized application of these models is a practical one. As we have discussed shortly in Section 2.7, some factors or dimensions of a case might relate to abstract facts or concepts that cannot be reliably determined directly. However, often these abstract concepts can be reduced to more base level concepts, which are more easily determined by the facts. Extending the reasoning to these intermediate determinations reduces the complexity of obtaining a fact situation.

From the legal context, we want to discuss two possible ways of thinking about these hierarchies that have been discussed in recent publications. Bench-Capon and Atkinson argue for the importance of *issues* [BCA21], while Horty and Canavotto stress the importance of *intermediate factors* [CH23b]. While the debate in the research is dominated by the connection of the model to legal practice, we can still find some interesting aspects of each viewpoint for general precedent reasoning.

Let us begin with Bench-Capon and Atkinson, and issues[1]. An issue in the context of a legal case is a question that is answered during the trial based on the facts, with the outcome of the trial depending on the answers to possibly many issues. The example

[1] The concept of an issue in a legal case is of course older than the work we are discussing, we are citing Bench-Capon and Atkinson because of them arguing for incorporating issues in formal precedential reasoning models.

in [BCA21] comes from the domain of trade secret law, with the central question of each case being: Did the defendant steal trade secrets from the plaintiff? An issue in this example could then be, whether some information actually is a trade secret, and whether the method of obtaining it was actually illegal. These issues are then combined in logical rules to determine the outcome that are, as opposed to the rules of the reason model, **not** defeasible. The reasoning therefore disconnects the factors from the outcome of the case, by using the factors to determine issues (using precedent and defeasible rules) and then using the issues to determine the outcome purely deductively. One important consequence of this separation is, that this essentially groups the factors by which issues they relate to. This means, that if all the factors relating to the issue of information being a trade secret are identical, then precedent will constrain the decision of this issue, even if other factors not related to the issue are different. In essence, using issues allows to apply precedent only for certain parts of a case. In the single-step reason model, any factor might be used to distinguish a precedent case.

Horty and Canavotto accept the notion of issues and their role in deciding case outcomes purely deductively, however they argue that determining the issues in a single step is not ideal, as there are factors that do not arise to the significance of an issue, but are still too complex to just be determined by the facts. Incorporating these intermediate factors and having them be subject to precedential constraint, they argue, offers meaningful information. They present the resulting model in [CH23a]. In this paper, they show that the two forms of constraint, constraint directly from base factors to issues (flat constraint) and constraint through a hierarchy of intermediate factors (hierarchical constraint) offer entirely different results, with there being cases that are constrained by flat constraint, but not by hierarchical constraint and vice versa. They argue that hierarchical constraint is a valuable direction to progress in, particularly when considering domains other than law. However, they do not offer a clear solution to the problem of cases being unconstrained by one model, but constrained by the other and vice versa.

Finally, we will take a closer look at the actual reasoning mechanisms of Horty and Canavotto's model with intermediate factors from [CH23a], and compare it with Prakken's model in [vWGPV23]. This model is a factor-based version of the full hierarchical model we briefly discussed in Section 2.7.

The first and main difference is that Prakken's model only applies a fortiori reasoning, and might thus be called a hierarchical result model, while Horty and Canavotto propose a hierarchical reason model. But even if one were to extend Prakken's approach to a reason model, there remains a difference in how the two models navigate the hierarchy. Prakken's hierarchical constraint is enforced recursively, with each step relying on the precedent that comes from recursive application. Starting at the base factors, which are provided in the fact situation, the model forces the decisions of intermediate factors based on the constraint mechanism, but does not include those factors in the fact situation. This then allows the constraint to be applied to higher level factors, and finally to the outcome. Deciding a case would then amount to querying the case base, whether there is any constraint on the decision of the outcome, which recursively looks for constraint of the factors involved in the decision. Importantly, the recursive constraint can lead to a

scenario, where a high-level factor is forced, without the lower level factors being actually determined. It suffices that they are constrained, which is then used in the recursive step. In Horty and Canavotto's model on the other hand, a decision of a case for an outcome is justified by an opinion, which always provides a full justification for every step. Such a judgment is built from the bottom up, with each intermediate factor being actually determined and included in the fact situation. Each individual step is therefore based on an actual fact situation in which all lower level factors have been determined.
Whether this difference is only conceptual, or has actual reasoning implications is however not clear, in particular because there is no reason model based version of Prakken's hierarchical model.

4.3 Incomplete case information

One of the strongest restrictions imposed by all the models we have discussed in detail, is that the fact situations are provided in full. In the factor-based case, a factor that is absent is purposefully absent, it is known that it is not present. In the dimensional case, each dimension must be assigned a value in the fact situation. This restriction is clearly a very strong one, and one that poses many practical problems. For example, there might be dimensions of a case that can be determined, but only at great cost. If such a dimension turns out to be irrelevant for a given case, then it might be wise to leave it undetermined and save the cost. In some cases, it might even be impossible to determine a dimension. Finding ways to still obtain constraint from these incomplete fact situations is therefore a vital step to enable the model to be applied in real-world AI systems. We will first summarize a short discussion by Prakken in [Pra21], which touches on some basics for the factor-based case. Then, we will discuss the work of Odekerken et al., specifically two papers in which they present a theory to handle incomplete fact situations [OBP23a, OBP23b] for the dimensional case, and mention a second development in the same direction [Rig24].
Beginning with the factor-based case, we want to discuss the absence of factors compared to the presence of a negated factor. To see better what this means, consider a fact situation of our running example, in which the applicant presents the two factors $smallApp$ and $prevPet$. We know that there is a factor $suffInc$ which applies if the applicant has sufficient income, however it is absent in the fact situation. In the model we discussed, the absence of $suffInc$ is interpreted as the applicant having insufficient income. If we were to include a factor for this, say $insuffInc$, would our fact situation of $smallApp, prevPet$ be the same as $smallApp, prevPet, insuffInc$? Since we so far assumed them to be the same it is interesting to investigate whether this assumption can be made without complications. In the words of knowledge-based systems, can we apply a closed-world assumption to our factors. For the factor-based result model, Prakken showed in [Pra21] that we can indeed do so, as the two fact situations would be subject to the same constraint, and would constrain other cases in the same way. For the reason model however, this no longer holds. Whether this result speaks for or against a closed-world assumption however is not part of the discussion, and definitely still allows for further

research. To the best of our knowledge, there has been no published work that includes notions like default negation into the factor based approach yet, to formally address the questions of open or closed-world assumptions.

The closest approach to using notions from default reasoning, or other techniques from formal logics, is that of Odekerken et al. [OBP23a] which defines concepts like stability, which have been developed in the past for example for incomplete knowledge bases.

In their work, the authors assume an incomplete fact situation using dimensions, where instead of each dimension being assigned a value, each dimension is assigned a nonempty set of possible values. They then define a notion of stability of such an incomplete fact situation, by checking whether all fact situations that are possible given the possible values are constrained in the same way. They also go on to define how a case base containing incomplete cases may constrain future cases.

Finally, Rigoni published another approach to dealing with incomplete dimensional fact situations, which uses the reason models form of constraint, as opposed to the result model in [OBP23a].

4.4 Explanations of the reason model

One final area of research that relates to the reason model is that of explanations. Explainable decisions are one of the most clear paths towards creating AI systems that can be trusted. As such, research into explainable AI has been growing in recent years, see for example the explainable AI conference XAI, which has just recently debuted [Lon23]. If the reason model or some generalization of it is to be used in AI contexts it is therefore crucial to investigate ways to explain the reason models decisions. For the basic factor-based reason model, there is already a way to provide explanations using the modal logic from [LLRS22]. In their work, the authors show how some common types of explanations for decisions of the reason model can be obtained. Their use of a logic for classifiers allows them to draw from the already established research of explanations for other AI systems, with classifiers being a well known concept. Some of the concepts they define for the reason model are the notion of prime implicants, minimal sets of factors that are sufficient for constraining a future decision, or abductive or constrastive explanations, two forms of showing for particular decisions why they are forced, or what would need to change for a different outcome to be permitted.

With the two new translations of reason models into classifier logic, a natural next step would be to investigate whether these approaches to explanations can also be applied to the two new translations.

CHAPTER 5

Conclusion and future work

In this thesis we have outlined and contributed to the current research objective of developing frameworks and models for the acquisition of normative knowledge for the use in AI applications. The motivation comes from the need for transparent and trustworthy systems, which can be addressed by symbolic representations of the normative information in some logic language. Acquiring and encoding this information into the chosen framework is currently done manually, which is infeasible at larger scale. A framework that uses a representation that can be more easily acquired and encoded would therefore be a valuable step forward. In the thesis we presented a candidate from the domain of legal reasoning in the form of the reason model for precedential reasoning and outlined four desirable aspects to combine into a model that takes a step towards general AI applications. These aspects are a rich representation of facts, an intuitive reasoning principle, a nuanced result and the ability to work in inconsistent case bases. All of these aspects have been present in some model in the literature, but there has never been a model that combines all aspects into one, which is what we have done in this work.

We began by presenting a series of models for precedential reasoning in Chapter 2. The first of these is the reason model by Horty [HB12], which can be seen as the starting point for many of the subsequent developments. It uses a basic fact representation in the form of binary factors, and uses a rule to justify the courts decision. It is the reason that makes up the premise of the rule, that gives the reason model its name and that is at the basis of the precedential constraint. Horty derives a preference of reasons based on each case, and defines constraint based on consistency within this preference relation.

The first modification we showed is a generalization of Canavotto [Canng] which lifts the requirement of the case base being consistent, and enables reasoning based on inconsistent case bases.

Following that we presented two modifications that replace the factor-based representation of case information with a more expressive dimension-based representation. These dimensions are sets of values that enable more precise descriptions of the facts of a case. Both the approaches we showed use reference values in their rules to justify the decision of the court, with Horty using reference values to build magnitude factors, which then allow the constraint to be defined as before, based on these special factors [Hor20]. The second approach of Prakken uses a more direct approach by providing rules that give a reference value for every dimension [Pra21]. These rules are then directly used to constrain decisions.

The last modification we presented is an adaptation of a model by Woerkom et al. which places the dimensions in a hierarchy [VGPV23]. The result of this approach is that the outcome of a single reasoning step is itself a dimension, which enables a many-valued outcome.

Finally, we highlighted two connections of the reason model to formal logics, first by presenting the deontic logic created by the constraint of the reason model, and second a modal logic encoding of the reason model, which can be used to obtain explanations for the decisions.

We then presented the contributions of this thesis, beginning with a formal comparison between the two dimension-based reason models we presented before. We formally define case base equivalence, and then show that transforming a case base that uses the complete rule model [Pra21] into a case base that uses the magnitude factor model [Hor20] results in equivalent case bases. Furthermore, we point out a flaw in the definition of the magnitude factor model, which relates to an earlier flaw of the model in the original version [Hor19].

These two circumstances lead us to choose an approach close to the complete rule model for our new model. We define the new model, which uses dimensions to represent the case facts, but also uses a dimension for the outcome. This accomplishes two of our desiderata, providing us with a rich representation of facts in the form of dimensions, and allows for nuanced results of the model by moving from two outcomes to a many-valued outcome. We also introduced the option to provide a case ordering as an outcome, allowing for direct comparison of cases without the need for any concrete values.
To accomplish the goal of an intuitive reasoning mechanism, we use the concept of rules as they have been used in the previous reason models. In order to maintain the concept of a rule that is used to justify a decision only towards one outcome, our model uses dedicated lower and upper bound rules to justify the decision. Furthermore, we achieve the final desired feature by generalizing our model to enable reasoning based on inconsistent case bases following the intuitions of Canavotto's generalization of the original factor based reason model [Canng].

On top of the definition of the new model, we also provide translations of the generalized reason model of [Canng] and of the complete rule model [Pra21] into BCL, following the work of [LLRS22]. These two translations can be seen as stepping stones towards an eventual translation of the new model we defined into BCL, which would open the model up for investigations into explanations.

In Chapter 4 we discussed other research areas related to precedential reasoning models and to the application for normative knowledge acquisition. These subjects outline the ideas and directions that the model might get developed towards in the coming years. Particularly interesting is the work into incomplete fact situations [OBP23a], as well as explanations of the models decisions using a modal logic encoding [LLRS22]. Both of these aspects are necessary features to make the model useful in practical AI applications.

We close this thesis with a discussion of some other future areas of research.

5.1 Future work

Aside from the continuations of the work we have outlined thus far, including incomplete fact situations and developing more translations of reason models into a formal logic, there are a few aspects that are relevant when we want to consider moving the model fully into an AI context.
The first of these aspects relates to the normative information that the model uses. All models we discussed in this work rely entirely on learned information, with every norm that is established in the system being derived from a case. For the domain of common law, this might be an acceptable setting, but for general AI applications, it might be useful to have some fixed norms as part of the initial system. Adapting the current models into mixed models, which are given an initial set of norms that are then adapted and expanded upon using the concepts of case-based information is a promising research direction. Possible directions might simply be to include statutory preferences in the preference relation of the factor based reason model, or to include rules that are not connected to some case as fixed rules.

This topic leads directly into the question of the context the reason model will be placed in. We presented the model as a way of acquiring and reasoning with normative information, however for more complex normative reasoning, there might be a need for additional systems that incorporate norms obtained by the reason model. The reason model might be placed in a larger framework as a way to obtain and process normative information that is then passed along with other norms to a more versatile system that uses the norms in its decision making process. It is in this larger context that the deontic logic also becomes more relevant. The deontic logic we presented in Section 2.8.1 is very simply and does not offer much use aside from a different perspective on the constraint of the reason model. However, a model with a many-valued outcome like our new model could be used as the basis of a deontic logic that uses graded deontic operators which could be used for example to resolve conflicts between norms.

Finally, acquiring the case information automatically would be a huge step towards fulfilling the promise of a full hybrid approach towards normative AI systems like we outlined in Chapter 1. In our examples, specifically in the scenario of our running example, we outlined how the case information may be obtained by gathering questionnaires filled with hypothetical cases. Ideally, the system would also be able to learn case information in a more natural way, for example by using large language models like GPT-4 or BERT [DCLT18], to read text-based case descriptions, and extract fact situations. While this does not seem feasible with the current tools, even more capable systems might offer this opportunity in the future. Some existing and promising research into the subject comes from automated legal reasoning. Using machine learning to process legal documents has seen high interest recently, as can be seen in the proceedings of the JURIX conference of 2023 [SSvD23]. There are also other approaches that instead tie into research on classifiers, for example developing the approach in [DFLL+23], where the authors determine whether a feature constitutes a factor in the sense that it uniformly favors one side. While this thesis did not focus on this aspect of the long term goal, it is crucial to the actual realization, and so needs to be investigated further in future research.
We can see how there are promising steps towards the goal we outlined, as well as a variety of different approaches or directions that can be worked towards. In the rapidly developing field of AI, working on transparent and trustworthy systems that use verifiable reasoning principles and that incorporate normative information is a crucial part of developing sustainable systems that can benefit many aspects of our lives.

List of Figures

List of Tables

Bibliography

[AA97] Vincent Aleven and Kevin D. Ashley. Evaluating a learning environment for case-based argumentation skills. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law*, ICAIL '97, page 170–179, New York, NY, USA, 1997. Association for Computing Machinery.

[AAB15] Latifa Al-Abdulkarim, Katie Atkinson, and Trevor J. M. Bench-Capon. Factors, issues and values: revisiting reasoning with cases. In Ted Sichelman and Katie Atkinson, editors, *Proceedings of the 15th International Conference on Artificial Intelligence and Law, ICAIL 2015, San Diego, CA, USA, June 8-12, 2015*, pages 3–12. ACM, 2015.

[AAB16] Latifa Al-Abdulkarim, Katie Atkinson, and Trevor J. M. Bench-Capon. A methodology for designing systems to reason with legal cases using abstract dialectical frameworks. *Artif. Intell. Law*, 24(1):1–49, 2016.

[Åqv02] Lennart Åqvist. *Deontic Logic*, pages 147–264. Springer Netherlands, Dordrecht, 2002.

[BA17] Trevor J. M. Bench-Capon and Katie Atkinson. Dimensions and values for legal CBR. In Adam Z. Wyner and Giovanni Casini, editors, *Legal Knowledge and Information Systems - JURIX 2017: The Thirtieth Annual Conference, Luxembourg, 13-15 December 2017*, volume 302 of *Frontiers in Artificial Intelligence and Applications*, pages 27–32. IOS Press, 2017.

[BCA21] Trevor Bench-Capon and Katie Atkinson. Precedential constraint: the role of issues. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 12–21, New York, NY, USA, 2021. Association for Computing Machinery.

[Can22] Ilaria Canavotto. Precedential constraint derived from inconsistent case bases. In Enrico Francesconi, Georg Borges, and Christoph Sorge, editors, *Legal Knowledge and Information Systems - JURIX 2022: The Thirty-fifth Annual Conference, Saarbrücken, Germany, 14-16 December 2022*, volume 362 of *Frontiers in Artificial Intelligence and Applications*, pages 23–32. IOS Press, 2022.

[Canng] Ilaria Canavotto. Reasoning with inconsistent precedents. *Artificial Intelligence and Law*, pages 1–30, forthcoming.

[CH22] Ilaria Canavotto and John Horty. Piecemeal knowledge acquisition for computational normative reasoning. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '22, pages 171–180, New York, NY, USA, 2022. Association for Computing Machinery.

[CH23a] Ilaria Canavotto and John Horty. Reasoning with hierarchies of open-textured predicates. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, page 52–61, New York, NY, USA, 2023. Association for Computing Machinery.

[CH23b] Ilaria Canavotto and John F. Horty. The importance of intermediate factors. In Giovanni Sileno, Jerry Spanakis, and Gijs van Dijck, editors, *Legal Knowledge and Information Systems - JURIX 2023: The Thirty-sixth Annual Conference, Maastricht, The Netherlands, 18-20 December 2023*, volume 379 of *Frontiers in Artificial Intelligence and Applications*, pages 13–22. IOS Press, 2023.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DFLL+23] Cecilia Di Florio, Xinghan Liu, Emiliano Lorini, Antonino Rotolo, and Giovanni Sartor. Inferring new classifications in legal case-based reasoning. In *Legal Knowledge and Information Systems*, pages 23–32. IOS Press, 2023.

[HB12] John F. Horty and Trevor J. M. Bench-Capon. A factor-based definition of precedential constraint. *Artif. Intell. Law*, 20(2):181–214, 2012.

[Hor11] John Horty. Rules and reasons in the theory of precedent. *Legal Theory*, 17, 03 2011.

[Hor19] John Horty. Reasoning with dimensions and magnitudes. *Artificial Intelligence and Law*, 27, 09 2019.

[Hor20] John Horty. Modifying the reason model. *Artificial Intelligence and Law*, 29(2):271–285, 2020.

[JPN+20] Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3561–3562, New York, NY, USA, 2020. Association for Computing Machinery.

[LL23] Xinghan Liu and Emiliano Lorini. A unified logical framework for explanations in classifier systems. *Journal of Logic and Computation*, 33(2):485–515, 01 2023.

[LLRS22] Xinghan Liu, Emiliano Lorini, Antonino Rotolo, and Giovanni Sartor. Modelling and explaining legal case-based reasoners through classifiers. In Enrico Francesconi, Georg Borges, and Christoph Sorge, editors, *Legal Knowledge and Information Systems - JURIX 2022: The Thirty-fifth Annual Conference, Saarbrücken, Germany, 14-16 December 2022*, volume 362 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2022.

[Lon23] Luca Longo, editor. *Explainable Artificial Intelligence - First World Conference, xAI 2023, Lisbon, Portugal, July 26-28, 2023, Proceedings, Part I*, volume 1901 of *Communications in Computer and Information Science*. Springer, 2023.

[MLB+23] Intae Moon, Jaclyn LoPiccolo, Sylvan C Baca, Lynette M Sholl, Kenneth L Kehl, Michael J Hassett, David Liu, Deborah Schrag, and Alexander Gusev. Machine learning for genetics-based classification and treatment response prediction in cancer of unknown primary. *Nature Medicine*, 29(8):2057–2067, 2023.

[MMS+21] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021.

[OBP23a] Daphne Odekerken, Floris Bex, and Henry Prakken. Justification, stability and relevance for case-based reasoning with incomplete focus cases. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, page 177–186, New York, NY, USA, 2023. Association for Computing Machinery.

[OBP23b] Daphne Odekerken, Floris Bex, and Henry Prakken. Precedent-based reasoning with incomplete cases. In *International Conference on Legal Knowledge and Information Systems*, 2023.

[Pra21] Henry Prakken. A formal analysis of some factor- and precedent-based accounts of precedential constraint. *Artificial Intelligence and Law*, 29(4):559–585, 2021.

[RA87] E. L. Rissland and K. D. Ashley. A case-based system for trade secrets law. In *Proceedings of the 1st International Conference on Artificial Intelligence and Law*, ICAIL '87, page 60–66, New York, NY, USA, 1987. Association for Computing Machinery.

[RH16] Mark O. Riedl and Brent Harrison. Using stories to teach human values to artificial agents. In Blai Bonet, Sven Koenig, Benjamin Kuipers, Illah R.

Nourbakhsh, Stuart Russell, Moshe Y. Vardi, and Toby Walsh, editors, *AI, Ethics, and Society, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 13, 2016*, volume WS-16-02 of *AAAI Technical Report*. AAAI Press, 2016.

[Rig18] Adam Rigoni. Representing dimensions within the reason model of precedent. *Artificial Intelligence and Law*, 26, 03 2018.

[Rig24] Adam Rigoni. Toward representing interpretation in factor-based models of precedent. *Artificial Intelligence and Law*, pages 1–28, 2024.

[SSS17] Jaspreet Singh, Gurvinder Singh, and Rajinder Singh. Optimization of sentiment analysis using machine learning classifiers. *Human-centric Computing and information Sciences*, 7:1–12, 2017.

[SSvD23] Giovanni Sileno, Jerry Spanakis, and Gijs van Dijck, editors. *Legal Knowledge and Information Systems - JURIX 2023: The Thirty-sixth Annual Conference, Maastricht, The Netherlands, 18-20 December 2023*, volume 379 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2023.

[VGPV23] Wijnand Van Woerkom, Davide Grossi, Henry Prakken, and Bart Verheij. Hierarchical a fortiori reasoning with dimensions. In Giovanni Sileno, Jerry Spanakis, and Gijs van Dijck, editors, *Legal Knowledge and Information Systems - JURIX 2023*, Frontiers in Artificial Intelligence and Applications, pages 43–52. IOS Press, December 2023. Publisher Copyright: © 2023 The Authors.; 36th International Conference on Legal Knowledge and Information Systems, JURIX 2023 ; Conference date: 18-12-2023 Through 20-12-2023.

[vWGPV23] Wijnand van Woerkom, Davide Grossi, Henry Prakken, and Bart Verheij. Hierarchical precedential constraint. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, page 333–342, New York, NY, USA, 2023. Association for Computing Machinery.

[Wak18] John F Wakerly. *Digital design : principles and practices*. Pearson, NY, NY, fifth edition with verilog edition, 2018.